

SRIDHAR MAHADEVAN

CATEGORIES FOR AGI

VOLUME I OF THE CATEGORICAL AI RESEARCH PROGRAM

Copyright © 2026 Sridhar Mahadevan

PUBLISHED BY VOLUME I OF THE CATEGORICAL AI RESEARCH PROGRAM

TUFTE-LATEX.GOOGLECODE.COM

Licensed under the Apache License, Version 2.0 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0>. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

First printing, September 2026

SOME MATHEMATICIANS ARE BIRDS, OTHERS ARE FROGS. BIRDS FLY HIGH IN THE AIR AND SURVEY BROAD VISTAS OF MATHEMATICS OUT TO THE FAR HORIZON. THEY DELIGHT IN CONCEPTS THAT UNIFY OUR THINKING AND BRING TOGETHER DIVERSE PROBLEMS FROM DIFFERENT PARTS OF THE LANDSCAPE. FROGS LIVE IN THE MUD BELOW AND SEE ONLY THE FLOWERS THAT GROW NEARBY. THEY DELIGHT IN THE DETAILS OF PARTICULAR OBJECTS, AND THEY SOLVE PROBLEMS ONE AT A TIME.

FREEMAN DYSON, NOTICES OF THE AMS, VOL. 56, NO 2.

Contents

<i>Preface</i>	11
<i>A Trail Map of Categories for AGI</i>	17
<i>I A Categorical Language</i>	21
<i>Category Theory for AGI</i>	23
<i>Functors for AGI</i>	31
<i>Representable Functors and the Yoneda Lemma</i>	43
<i>Diagrams and Universal Constructions</i>	49
<i>II Compositional Learning</i>	57
<i>Categorical Deep Learning</i>	59
<i>Diagrammatic Backpropagation</i>	69
<i>III Geometric and Kan Extension Transformers</i>	77

<i>Geometric Transformers</i>	79
<i>Dynamic Compositionality</i>	95
<i>Information Regimes in Geometric Transformers</i>	127
<i>Kan Extension and Topological Coend Transformers</i>	137
<i>Structured Language Modeling</i>	177
<i>Manifold Learning with Geometric Transformers</i>	183
<i>Mean-Field Theory of Geometric Transformers</i>	191
<i>Depth Sweeps for Geometric Transformers</i>	199
<i>IV Categorical Models of Causality</i>	205
<i>Adjoint Functors</i>	207
<i>Causal Claims from Language</i>	213
<i>Temporal Diffusion over Causal Trajectories</i>	221
<i>Building Agentic Systems using Kan Extension Transformers</i>	243
<i>Topos Causal Models</i>	257
<i>Judo Calculus</i>	269

<i>CSQL: Mapping Documents into Topos Causal Model Databases</i>	305
<i>Homotopy in Language and Causal Inference</i>	343
<i>Model Categories for Causality and Language</i>	357
<i>Predictive State Representations in a Topos</i>	365
<i>Causal Density Functions</i>	397
<i>V Universal Decision Models</i>	407
<i>Universal Decisions with Kan Extensions</i>	409
<i>Universal Reinforcement Learning</i>	423
<i>Deep URL with Geometric Transformers</i>	427
<i>VI Frontiers of AGI</i>	435
<i>Consciousness</i>	437
<i>Universal Imitation Games</i>	441
<i>Formal Verification Map</i>	445
<i>CLIFF Companion</i>	451
<i>Code Companion</i>	459

Bibliography 463

Glossary of Notation 473

Subject Index 477

I propose to consider the question, "Can machines think?".

A. M. Turing (1950) , Mind 49: 433-460.

Preface

This book is an invitation to use category theory as a working language for artificial intelligence. It grew from *CMPSCI 692CT: Category Theory for AGI*, taught at the University of Massachusetts Amherst in Spring 2026, and from a broader conviction: many of the hardest problems in AI concern not an isolated model, but the relationships among models, representations, agents, data sources, memories, interventions, and decisions.

Traditional introductions to category theory are often written by mathematicians for mathematicians [Mac Lane, 1971]. Their abstraction is a strength, but it can leave an AI reader wondering where the concepts touch computation. This book takes the complementary route. It introduces categories, functors, natural transformations, the Yoneda lemma, universal constructions, adjunctions, monads, Kan extensions, sheaves, topoi, and coalgebras through problems that arise in contemporary learning and reasoning.

Its central claim is:

Intelligence is not well described as one input–output function. It is a compositional system of representations and transformations whose meaning depends on how its parts interact, transport information, and assemble into larger wholes. Category theory provides a language for studying that organization directly.

This claim is intentionally methodological. It does not say that a categorical description by itself makes a system intelligent, or that every AI problem requires advanced category theory. It says that when the scientific question concerns composition, equivalence, translation, context, local-to-global assembly, recursion, or universal structure, those ideas should be expressed in a mathematics where they are first-class.

Design principle

Study an AI component not only by its internal parameters, but also by the maps into and out of it, the transformations it preserves, and the larger diagrams in which it participates.

Why category theory?

Modern AI rests on linear algebra, calculus, graph theory, probability, statistics, and optimization. These tools remain indispensable. Linear algebra represents states and transformations; calculus describes local change; graphs encode connections; probability and statistics quantify uncertainty and evidence; optimization searches among candidate models. Category theory contributes a different level of description: it asks which constructions are preserved when these objects and operations are connected.

A graph records that two nodes are linked. A categorical diagram can also record the type and direction of each map, distinguish alternative composites, and declare that certain paths should agree. A set describes an object through its elements. A categorical description can characterize an object by its relationships or by a universal property: what it does in every compatible context. The Yoneda lemma makes this relational viewpoint precise, while limits, colimits, and Kan extensions describe canonical forms of assembly and extension.

This does not make category theory foundationally “richer” than set theory in a simple hierarchy; categorical constructions can be encoded in sets. The advantage is one of native expression and invariance. Membership is primary in set theory. Maps and composition are primary in category theory. The latter often lets us say what an AI construction means without tying that meaning to one coordinate system, data structure, neural implementation, or parameterization.

Mathematical language	Natural question	Representative AI role
Linear algebra and calculus	How is a representation transformed, and how does it change locally?	Embeddings, neural layers, gradients, and dynamics
Graph theory	Which variables, tokens, agents, or states are connected?	Architectures, causal graphs, relational data, and message passing
Probability and statistics	What is uncertain, supported, identifiable, or likely to generalize?	Inference, estimation, calibration, and evaluation
Optimization	Which candidate minimizes an objective or satisfies a constraint?	Training, planning, control, and model selection
Category theory	What composes, transports, extends, or glues, and which properties survive?	Interfaces, representation changes, modular systems, and local-to-global reasoning

Category theory organizes rather than replaces the familiar mathematics of AI.

Applied category theory has emphasized precisely this compositional role [Fong and Spivak, 2018, Spivak, 2014]. For AI, it offers three recurring benefits. First, it makes interfaces and compositions explicit. Second, it separates an abstract construction from its many realizations. Third, it provides universal objects and maps that explain why a construction is canonical rather than merely convenient.

Why AGI?

The phrase *artificial general intelligence* is used here as a horizon for system design, not as a claim that current systems have achieved a settled definition of general intelligence. The relevant challenge is breadth: perception, language, memory, causal reasoning, planning, learning, interaction, and decision making must coexist and remain mutually intelligible. A narrow model may be evaluated in isolation. A general system must compose capabilities across tasks, contexts, time scales, and agents.

Turing’s imitation game placed interaction at the center of machine intelligence [Turing, 1950]. Category theory strengthens that relational perspective. An object can be studied through the family of ways in which other objects probe it; an agent can be studied through its interactions and transformations; and a complex architecture can be studied through the commuting diagrams and universal constructions that organize its parts.

The title therefore signals the scale of the questions rather than promising a categorical recipe for AGI. The book asks what category theory can clarify about systems that learn representations, compose modules, reason across contexts, infer causal structure, maintain predictive state, choose actions, and interact with humans.

Boundary

Category theory supplies a language of structure and composition. It does not replace empirical evidence, statistical validation, computational complexity, safety analysis, or domain knowledge. A beautiful universal construction can still be the wrong model of the world.

The path through the book

The book is organized into six sections. Their sequence is a conceptual path, not a claim that every reader must traverse all six in order.

SECTION I ESTABLISHES A CATEGORICAL LANGUAGE. Categories, morphisms, functors, natural transformations, representables, the Yoneda lemma, diagrams, limits, and colimits provide the common vocabulary.

SECTION II TURNS COMPOSITION INTO A LEARNING QUESTION. Categorical deep learning and diagrammatic backpropagation ask how declared relations among components can influence credit assignment rather than merely describe a trained system after the fact.

SECTION III STUDIES GEOMETRIC AND UNIVERSAL ARCHITECTURES. Geometric Transformers, dynamic compositionality, information regimes, Kan extensions, structured language models, manifold learning,

mean-field analysis, and scaling laws connect categorical constructions to trainable models and experiments. The purpose is not categorical renaming: each construction must expose an architectural choice, invariant, or testable consequence.

SECTION IV DEVELOPS CATEGORICAL MODELS OF CAUSALITY.

Adjoint functors prepare the language of transport. The subsequent chapters study causal extraction from language, temporal trajectories, agentic workflows, topos causal models, j -do calculus, causal databases, homotopy, model categories, predictive states, and causal densities. A structural transport acquires causal meaning only when its observational and interventional semantics are supplied.

SECTION V TURNS TO UNIVERSAL DECISION MODELS. Kan-extended decisions and coalgebraic reinforcement learning organize observation, state, action, and future behavior through time.

SECTION VI MARKS THE FRONTIER AND RECORDS THE COMPANIONS. Consciousness and generalized imitation games are exploratory applications, not settled theories of mind. The formal-verification, CLIFF, and code companions separately record machine-checked mathematics, interactive software, and executable demonstrations. These forms of support complement one another but are not interchangeable.

How to read the book

No prior category theory is assumed. Readers should be comfortable with basic linear algebra, multivariate calculus, probability and statistics, graphs, and the central ideas of machine learning. Programming examples use Python, and the computational companion provides notebooks suitable for local use or Google Colab.

The book supports several routes:

1. A first-time reader can follow the opening chapters in order through diagrammatic backpropagation, acquiring the categorical vocabulary before choosing an application track.
2. A deep-learning reader can continue through the geometric and Kan-extension chapters, returning to adjunctions when the applications demand them.
3. A causal-inference reader can move from diagrams and universal constructions to adjoints, causal language, topoi, sheaves, homotopy, and predictive-state representations.

4. A reinforcement-learning or agent-systems reader can use the early functor and coalgebra material as preparation for universal decisions, coalgebraic RL, consciousness, and imitation games.

Exercises are invitations to reconstruct the ideas rather than tests of symbol manipulation alone. The notebooks provide small executable examples; the Lean development records which mathematical statements have been machine-checked. Neither companion should be confused with the other: successful code is not a proof, and a proof of a structural theorem is not an empirical validation of an AI system.

The boundary of the book

This is neither a comprehensive category-theory text nor a survey of all artificial intelligence. It is an introductory, application-centered book shaped by the author's research program. Some chapters establish standard mathematics; others develop recent architectures and experiments whose scope and limitations must be judged individually.

The book also does not claim that abstraction is automatically explanatory. A categorical reformulation earns its place only when it clarifies an interface, exposes an invariant, identifies a universal construction, organizes local and global information, or supports a comparison that was previously obscure.

The durable question is not whether category theory can be placed around an AI system, but whether doing so reveals what the system preserves, how its parts compose, and which conclusions survive a change of representation.

That question guides the chapters that follow.

A Trail Map of Categories for AGI

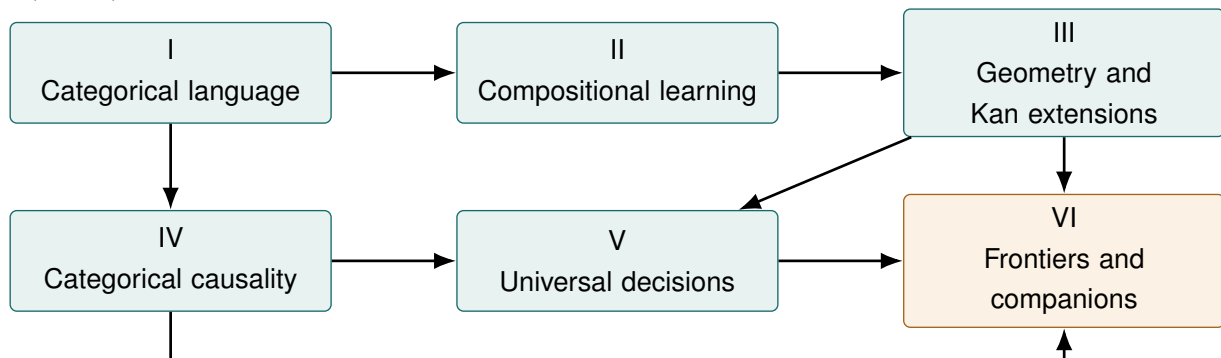
The Preface explains why category theory is a useful working language for artificial intelligence. This trail map shows how the book's many subjects fit together before the detailed journey begins. Unlike a text organized around one algorithm or one formal calculus, this book surveys a landscape: categorical language, compositional learning, geometric architectures, causality, decisions, and open questions about general intelligence.

Design principle

Read each construction at three scales: as a mathematical object, as a possible architecture or algorithm, and as a claim that requires its own formal or empirical support. The maps between these scales are part of the subject.

The landscape from above

The six sections are connected by a common question: what becomes visible when objects are studied through their maps, compositions, and changes of context? The opening language makes this question precise. Compositional learning and geometric transformers turn it into trainable computation. Categorical causality asks when structural transport supports observational, interventional, or local-to-global reasoning. Universal decision models add state, action, and time. The final section tests how far these ideas can reach.



The arrows mark conceptual dependencies, not a mandatory reading order. Later regions frequently send the reader back to earlier universal constructions.

Six regions of the book

Each section enlarges the range of systems that can be described, but it also introduces new obligations. The following table records the principal destination of each region rather than attempting to summarize every result.

Region	Principal constructions	Question carried forward
I. A Categorical Language <i>four opening chapters</i>	categories, morphisms, functors, natural transformations, representables, Yoneda, diagrams, limits, and colimits	how can an AI system be characterized by its relations and universal properties rather than by one presentation?
II. Compositional Learning <i>two chapters</i>	categorical deep learning and diagrammatic backpropagation	how can declared compositional structure participate in learning and credit assignment?
III. Geometric and Kan Extension Transformers <i>eight chapters</i>	geometric transformers, dynamic compositionality, Kan extensions, coends, structured language models, manifolds, mean-field theory, and scaling laws	which geometric and universal constructions can be realized in trainable architectures, and what do experiments establish?
IV. Categorical Models of Causality <i>eleven chapters</i>	adjunctions, causal language, temporal diffusion, agentic systems, topoi, j -do calculus, causal databases, homotopy, model categories, predictive states, and causal densities	when do transport, gluing, and comparison acquire causal meaning rather than only structural meaning?
V. Universal Decision Models <i>three chapters</i>	Kan-extended decisions, coalgebraic reinforcement learning, and deep universal reinforcement learning	how should observations, state, actions, and future behavior compose through time?
VI. Frontiers of AGI <i>five chapters</i>	consciousness, universal imitation games, formal verification, the CLIFF interface, and executable code	which claims are definitions, theorems, implementations, experiments, or open conjectures?

A map, not a ladder

The sections increase in specialization, not in categorical “height.” A reader interested in causality or reinforcement learning may enter those regions after learning the required vocabulary and return to earlier chapters as adjunctions, Kan extensions, sheaves, topoi, or coalgebras become necessary.

The trails that cross the regions

The book coheres through recurring constructions rather than through a single end-to-end algorithm. Five trails are especially useful for recognizing when an idea has reappeared in a new guise.

Recurring trail	Main route through the sections	What changes along the route
Composition and coherence	$I \rightarrow II \rightarrow III \rightarrow IV \rightarrow V$	path equality becomes a learning obligation, an architectural constraint, a local-to-global condition, and finally a condition on decisions through time
Representation and translation	$I \rightarrow III \rightarrow IV$	functors and natural transformations become representation maps, language translations, and comparisons between observational or causal contexts
Universal extension	$I \rightarrow III \rightarrow IV \rightarrow V$	limits, colimits, adjunctions, and Kan extensions move from definitions to architectural extension, causal transport, and decision synthesis
Geometry and local-to-global assembly	$I \rightarrow III \rightarrow IV$	diagrams and representables lead to manifolds, sheaves, topoi, homotopy, and the problem of gluing compatible local evidence
Dynamics, state, and agency	$IV \rightarrow V \rightarrow VI$	temporal causal trajectories and predictive states become coalgebraic policies, evolving agents, and interaction-based accounts of intelligence

Five kinds of claim

The book deliberately moves among several epistemic levels. Keeping them distinct is part of reading the book well.

Standard mathematics. A definition or theorem records established categorical structure, with references to the mathematical literature.

Proposed construction. A new architecture, semantics, or categorical model states what is being proposed and which properties have been derived.

Computational realization. An algorithm, database, interface, or notebook shows how part of a construction is represented and executed.

Empirical realization. A benchmark, ablation, scaling study, or case analysis tests a stated consequence under particular experimental conditions.

Frontier interpretation. A conjectural application—especially in the chapters on consciousness and imitation—organizes questions without claiming that the underlying phenomenon has been explained.

These forms of support can reinforce one another, but they are not interchangeable. A universal property does not certify an implementation; runnable code does not prove a theorem; an empirical improvement does not uniquely establish its proposed explanation; and a mathematically coherent frontier model may remain scientifically speculative.

Choose a route

The opening four chapters provide the common vocabulary. After that point, readers can choose a trail and return to prerequisites as needed.

Interest	Suggested route	Guiding question
First course in categorical AI	Categorical Language → Compositional Learning, then selected chapters from each remaining section	which categorical constructions most directly clarify familiar AI systems?
Deep learning architectures	functors and diagrams → Compositional Learning → Geometric and Kan Extension Transformers	when does categorical structure alter an architecture, training signal, or empirical prediction?
Causality and scientific inference	diagrams and universal constructions → adjoints → Categorical Models of Causality	when can local evidence, transport, or comparison support a causal claim?
Decisions, reinforcement learning, and agents	functors and universal constructions → predictive states → Universal Decision Models → imitation games	how should state, observation, action, and interaction be organized through time?
Formal and computational companions	study a substantive chapter together with its Formal Verification Map, CLIFF, and Code Companion entries	which part of the claim has been formalized, implemented, or empirically tested?

Signposts for the journey

1. **A categorical description is not automatically an explanation.** It earns explanatory force by exposing an invariant, universal property, compositional constraint, or comparison that matters to the AI problem.
2. **Shared syntax does not guarantee shared semantics.** A Kan extension, adjunction, sheaf, or coalgebra must be interpreted in its domain; the formal construction alone does not supply causal, cognitive, or decision meaning.
3. **Levels of support must remain visible.** Standard mathematics, new theory, software, and empirical demonstrations answer different questions and should be evaluated by different criteria.

With these signposts, the size of the book becomes an advantage. Readers can move between the bird’s-eye view of universal structure and the frog’s-eye details of architectures, datasets, proofs, and experiments, while keeping track of which conceptual trail connects them.

The three-book sequence. This volume supplies the broad categorical language. *Machine Learning from Enforcing Compositionality* develops typed structural diagnosis and repair. *Infinitesimal Creativity* studies finite extension of a declaration. The later volumes require their local prerequisites, not every exploratory chapter here.

Section I

A Categorical Language

Category Theory for AGI

ARTIFICIAL GENERAL INTELLIGENCE names an unsettled ambition: to build systems whose competence extends across perception, language, memory, reasoning, learning, and action rather than remaining confined to one task. The phrase does not yet identify a single mathematical object or an agreed empirical threshold. In this book it functions as a design horizon. The question is what mathematical language can help us reason about many capabilities that must interact without losing their individual semantics.

TURING’S IMITATION GAME replaced the unanswerable question “Can machines think?” with an operational test based on linguistic interaction [Turing, 1950]. Its importance here is relational: competence is judged through an organized exchange between a system and an observer, not by inspection of an alleged inner essence. The imitation game is neither a complete definition of intelligence nor a settled test for AGI. It is an early example of characterizing a system through the morphisms—questions, answers, and discriminations—that connect it to other systems.

Accordingly, this book treats human-like interaction as one probe of a more general compositional problem. An AGI system would have to coordinate many forms of competence across contexts and timescales; passing one behavioral test would not by itself establish that broader capacity.

A Frontier Question: Consciousness

Consciousness is one of the book’s frontier applications. Human cognition appears to coordinate a narrow field of deliberate attention with many parallel processes that operate outside immediate awareness. Skilled action offers a familiar illustration: a novice pianist attends to individual movements that an expert performs with little conscious effort. These observations motivate questions about memory, dynamics, integration, and agency, but they do not determine a unique mathematical theory.

The figure above records one categorical proposal [Mahadevan, 2025a]. It combines internal languages in topos theory, coalgebraic dynamics, and multi-agent decision structures. The relevant mathematical tools are developed throughout the book; the proposed interpretation is examined only near

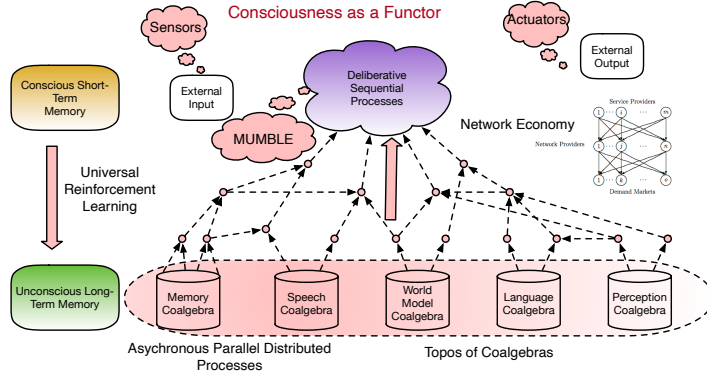
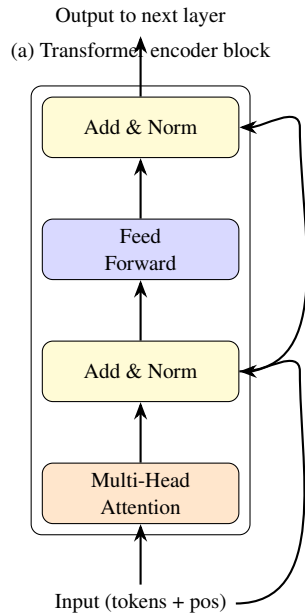


Figure 1: A proposed categorical architecture for consciousness. The diagram is a modeling hypothesis revisited in Section VI, not a theorem that category theory explains consciousness.

the end. Beginning with this frontier makes the ambition visible while keeping its epistemic status explicit. We now turn to the more concrete example that drives much of the book: Transformer architectures.

Transformers

GENERATIVE PRE-TRAINED TRANSFORMERS (GPTs) are among the most capable contemporary AI systems. This book does not treat their capabilities as evidence that AGI has been achieved. It uses the Transformer architecture [Vaswani et al., 2017] as a recurring example of a system whose modules, residual paths, representations, and contexts must compose. The standard encoder block can be visualized by the following diagram.



TRANSFORMER ARCHITECTURES can be studied at several levels. At the implementation level they are deep neural networks; at the language-model level they are sequence processors; and at the categorical level they are composites of typed transformations whose ordering and compatibility may matter [Bengio, 2009, Xiao and Zhu, 2023]. The last viewpoint is the distinctive concern of this book.

For example, a PostLN block places layer normalization after the attention or feed-forward sublayer, whereas a PreLN block changes that order. The two arrangements contain closely related components but have different training behavior [Kim et al., 2025]. This is a small instance of the book’s larger question: when does the way components compose change what a system can learn or how reliably it can be trained?

From Transformers to Function Categories

THE CATEGORICAL STARTING POINT is modest. A trained Transformer implements a parameterized map between spaces of token representations. We first place such maps inside well-defined categories of functions. Only then do we ask which larger ambient categories have topos structure. This order prevents a useful analogy from being mistaken for a theorem about neural networks.

Definition 1. A category $\mathcal{C}$ consists of objects; a collection $\mathcal{C}(c, d)$ of morphisms for every ordered pair of objects; an identity $1_c \in \mathcal{C}(c, c)$ for every object; and composition maps

$$\mathcal{C}(c, d) \times \mathcal{C}(d, e) \longrightarrow \mathcal{C}(c, e).$$

Composition is associative, and the identity morphisms are left and right units. Objects may represent sets, vector spaces, data types, model components, or entire systems, but the morphisms and their composition must be specified rather than inferred from the names of the objects.

When you read this definition, you should try to construct a few examples of categories. An *initial object* c in category $\mathcal{C}$ is defined as one inducing a unique arrow from c to every object in category $\mathcal{C}$. A *terminal object*, usually denoted by $\mathbf{1}$, is one that defines a unique arrow from every object c in category $\mathcal{C}$ into $\mathbf{1}$. Not all categories have initial or final objects.¹ An object c is isomorphic to another object d , denoted $c \simeq d$, if two arrows $f : c \rightarrow d$ and $g : d \rightarrow c$ exist, such that $g \circ f = 1_c$, and $f \circ g = 1_d$.

A word on notation: $\mathcal{C}(a, b)$ and $\mathbf{Hom}_{\mathcal{C}}(a, b)$ both denote the morphisms from a to b . An ordinary locally small category treats this hom-object as a set. In an *enriched* category, hom-objects may instead live in a specified monoidal category—for example, they may be vector spaces. Some AI applications benefit from enrichment, but it is additional structure and will be invoked only when the relevant composition law has been defined.

¹ Exercise: Consider the category of **Sets**. Does it have a terminal object? An initial object? What are they, if they exist?

WHERE TRANSFORMERS ENTER. A trained Transformer is a concrete realizable approximation to a sequence-to-sequence function $f_\theta : \mathbb{R}^{n \times d} \rightarrow \mathbb{R}^{n \times d}$. Thus, we can regard Transformer models as inhabiting (or approximating) arrows in a function-based topos-like universe. This viewpoint is not a claim that “the set of all neural networks” is itself an elementary topos without further structure; rather it is a guiding semantics that becomes operational only after patch sites, sheaves, and internal languages have been specified.

TOPOS theory can give us a powerful guide to designing a novel class of Transformer models². To give some intuition, let us consider a simple example of a topos category, where every object is simply a function $f : A \rightarrow B$, from some arbitrary domain set A to some co-domain set B . To define a category of functions on sets, we need to specify both the objects of the category – this is straightforward, as it is simply every possible function – as well as the collection of arrows between every pair of objects. This appears less straightforward: how do we define arrows between function objects?

² Sridhar Mahadevan. Topos theory for generative AI and LLMs, 2025f. URL <https://arxiv.org/abs/2508.08293>

COMMUTATIVE DIAGRAMS make compatibility explicit. Like directed graphs, they have objects and arrows, but they additionally assert that specified composites agree. In the arrow category of sets, a morphism between two function objects is represented by the following square:

$$\begin{array}{ccc} A & \xrightarrow{h} & A' \\ \downarrow f & & \downarrow f' \\ B & \xrightarrow{g} & B' \end{array}$$

Here, A, A', B and B' denote sets, and f, f', g and h define functions between a pair of sets as shown in the diagram. This diagram should be viewed as defining a set of constraints. It asserts that the *composition* of functions $g \circ f$ (meaning, first apply f and then apply g) is equivalent to the composition $f' \circ h$. This commutativity condition is exactly the condition for a morphism between the objects f and f' in the arrow category.

A PRECISE TOY TOPOS. There is a mathematically clean way to package “functions as objects” into an elementary topos: if $\mathcal{E}$ is an elementary topos (e.g., **Set**), then its *arrow category* $\mathcal{E}^{[1]}$ —whose objects are arrows $f : A \rightarrow B$ and whose morphisms are commutative squares—is again an elementary topos. We will use **Set**^[1] as our running toy model.

YONEDA EMBEDDINGS. Chapter 3 introduces the Yoneda embedding

$$y : \mathcal{C} \longrightarrow \mathbf{Set}^{\mathcal{C}^{op}}, \quad c \longmapsto \mathcal{C}(-, c).$$

For a small category $\mathcal{C}$, the codomain is a presheaf topos. Yoneda does not say that $\mathcal{C}$ itself is a topos; it embeds $\mathcal{C}$ fully faithfully into a canonical topos

of set-valued presheaves [Mac Lane and Moerdijk, 1992]. This distinction between an object and a richer ambient semantics is precisely the distinction being used for neural maps.

MORPHISMS BETWEEN FUNCTION-OBJECTS. In the arrow category $\mathbf{Set}^{[1]}$, a morphism from $f : A \rightarrow B$ to $f' : A' \rightarrow B'$ is precisely a pair of functions $h : A \rightarrow A'$ and $g : B \rightarrow B'$ such that $g \circ f = f' \circ h$. That is, morphisms are commutative squares.

In the arrow topos $\mathbf{Set}^{[1]}$, the objects are all functions between sets and the morphisms are the commutative squares just described. A trained Transformer may realize or approximate one such function, but the collection of realizable Transformer maps is only a selected class of objects. It is not automatically closed under the limits, exponentials, or subobject classifiers of the ambient topos.

TOPOS STRUCTURE BELONGS FIRST TO THE AMBIENT CATEGORY. The universal-approximation theorem for Transformers [Yun et al., 2020] concerns approximation of certain sequence-to-sequence functions. It does not prove that the class of Transformer architectures is itself an elementary topos. The rigorous construction used here is the arrow topos $\mathbf{Set}^{[1]}$; neural models enter as implementations of, or approximations to, some of its arrows.

To understand why a topos is a useful ambient semantics, we introduce its defining structure in the clean setting of sets and function objects. A central property is the ability to classify subobjects categorically, without depending on a particular representation of their elements.

To help build some intuition, consider how to define subsets without “looking inside” a set. Essentially, a subset S of some larger set T can be viewed as a “monic arrow” (an injective function). Monic arrows generalize injective functions. An arrow $f : a \rightarrow b$ in a category $\mathcal{C}$ is called **monic** if given any parallel arrows $g, h : c \rightrightarrows a$, the equality $f \circ g = f \circ h$ implies that $g = h$, namely f is “left-cancellable”. Now,

Definition 2. In a category $\mathcal{C}$, a **subobject classifier** is a $\mathcal{C}$ -object Ω , and a $\mathcal{C}$ -arrow $\mathbf{true} : \mathbf{1} \rightarrow \Omega$, such that to every monic arrow $S \hookrightarrow X$ in $\mathcal{C}$, there is a unique arrow ϕ that forms the following pullback square commutative diagram:

$$\begin{array}{ccc} S & \xrightarrow{\quad} & \mathbf{1} \\ \downarrow m & & \downarrow \mathbf{true} \\ X & \xrightarrow{\quad \phi \quad} & \Omega \end{array}$$

What does this diagram mean? Consider the category of **Sets**. Let S be some subset of X . Then, m defines a monic (1-1) function that simply maps each element $x \in S$ to the same element $x \in X$. ϕ is the characteristic (Boolean) function that maps X to a Boolean $\{0, 1\}$ -valued object Ω . The

arrow **true** simply maps **1** to the element 1 in Ω . The diagram commutes, hence for each element $x \in X$, it must be the case that composing the **true** with the unique function mapping S to **1** (which recall is a terminal object) gives us the exact same arrow as composing ϕ with m .

This pullback says that every subobject is recovered by pulling back the generic truth arrow along a unique characteristic map. In **Set**, $\Omega = \{0, 1\}$. In a general topos, the truth values form a Heyting algebra and the internal logic is intuitionistic. Any application to neural models must therefore specify an ambient topos and explain what its subobjects and characteristic maps mean operationally.

The cited topos formulation of generative models [Mahadevan, 2025f] develops one proposed semantics along these lines. The proposal should be distinguished from the standard theorem that an arrow category of a topos is again a topos.

Definition 3. A category $\mathcal{C}$ has **binary products** if for every pair of objects, c and d , there exists a third object, $e \simeq c \times d$, along with two projection arrows, $p_1 : e \rightarrow c$ and $p_2 : e \rightarrow d$, such that for any other object a and arrows $f : a \rightarrow c$ and $g : a \rightarrow d$, there exists a unique morphism $u : a \rightarrow e$ satisfying $p_1 \circ u = f$ and $p_2 \circ u = g$.

This abstract definition is a lot like many others you will encounter in this book. So, it takes some patience to work through what it means.³ Products are guaranteed in the ambient arrow topos. Whether a restricted class of implemented Transformer maps is closed under the corresponding construction is a separate architectural question, not something supplied by terminology.

³ Exercise: draw the commutative diagram for this definition.

Definition 4. A category $\mathcal{C}$ with binary products has **exponential objects** if for each pair of objects c, d in $\mathcal{C}$, there exists an object c^d that defines the following bijection:

$$\mathcal{C}(e \times d, c) \simeq \mathcal{C}(e, c^d)$$

EXAMPLE (SETS). In **Set**, the exponential object Y^X can be taken to be the set of all functions $X \rightarrow Y$. There is a canonical *evaluation* map

$$\text{ev} : Y^X \times X \rightarrow Y, \quad \text{ev}(f, x) = f(x),$$

and the universal property of Y^X states that every function $F : E \times X \rightarrow Y$ corresponds uniquely to its curried form $\tilde{F} : E \rightarrow Y^X$. We will repeatedly reuse this pattern when we interpret modules inside Transformer architectures as “function objects” and later when we interpret patchwise reasoning as internal higher-order structure.

Definition 5. A category $\mathcal{C}$ is **Cartesian closed** if it has binary products, a terminal object **1**, and exponential objects.

These definitions specify the ambient structure that the later applications use. They do not imply that Transformer models, considered in isolation, form a topos.

Definition 6. An **elementary topos** is a category $\mathcal{C}$ with finite limits, exponential objects, and a subobject classifier. Equivalently, it is a finitely complete Cartesian closed category with a subobject classifier. Cartesian closedness by itself supplies finite products but does not, by definition, supply equalizers; the finite-limit hypothesis must therefore be stated separately [Mac Lane and Moerdijk, 1992, Chapter IV].

For example, **Set** is a topos. Products are Cartesian products, the exponential Y^X is the set of functions from X to Y , and a singleton is terminal. The two-element set $\Omega = \{\text{false}, \text{true}\}$, equipped with the map selecting true, is a subobject classifier: characteristic functions classify subsets by pullback.

THE TABLE below gives a first dictionary from familiar set language to its categorical generalization. The entries are analogies, not definitions; the universal properties introduced here supply the precise meaning.

Set theory	Topos theory
set	object
subset	subobject
truth values $\{0, 1\}$	subobject classifier Ω
power set $P(A) = 2^A$	power object $P(A) = \Omega^A$
bijection	isomorphism
injection	monic arrow
surjection	epic arrow
singleton set $\{*\}$	terminal object 1
empty set $\emptyset$	initial object 0
elements of a set X	morphism $f : \mathbf{1} \rightarrow X$
-	functors, natural transformations
-	limits, colimits, adjunctions

Summary

This chapter has established the book's first reader contract. AGI is a design horizon rather than a settled mathematical definition. Turing's imitation game supplies an influential relational probe, while consciousness remains a frontier modeling problem. Transformer networks provide the concrete running example: their components can be represented as typed maps whose ordering and compatibility deserve explicit study.

The rigorous categorical claim is about the ambient mathematics. The arrow category **Set**^[1] is a topos whose objects are functions and whose morphisms are commutative squares. A Transformer can implement or approximate selected maps in such a setting; this does not make the restricted class of Transformer models a topos. That distinction between ambient semantics and realizable architecture will recur throughout the book.

THE COMPUTATIONAL COMPANIONS support a different layer of the story. Public Transformer implementations provide concrete architectural examples, while CLIFF connects passages in the book to guided questions, demonstrations, and code. Neither implementation establishes a categorical theorem, and neither theorem substitutes for empirical evaluation. Chapter 2 now develops functors, the maps that let structure move between categories.

Suggested Reading

For systematic introductions to categories, functors, universal properties, and adjunctions, begin with [Mac Lane \[1971\]](#) or [Riehl \[2017\]](#). [Spivak \[2014\]](#) develops the subject for scientific applications, while [Fong and Spivak \[2018\]](#) emphasizes compositional modeling through concrete examples.

Functors for AGI

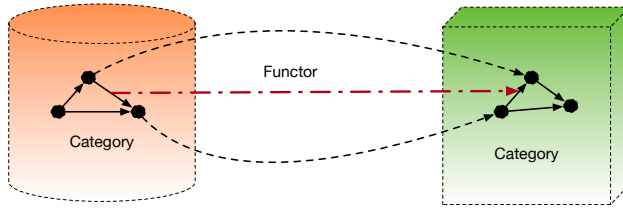


Figure 2: Functors act on both the objects and arrows of a category.

FUNCTORS translate one category into another while preserving identities and composition. This chapter develops that definition, introduces natural transformations as morphisms between functors, and then previews two application patterns used later in the book: functorial algorithms and coalgebraic descriptions of state-based systems. The categorical definitions are standard; the AI examples are modeling choices whose validity depends on the categories and morphisms being specified explicitly. Section 1.3 of Riehl’s textbook gives a careful introduction [Riehl, 2017].

Definition 7. A functor $F : \mathcal{C} \rightarrow \mathcal{D}$ between categories $\mathcal{C}$ and $\mathcal{D}$ assigns an object Fc of $\mathcal{D}$ to each object c of $\mathcal{C}$ and a morphism $Ff : Fc \rightarrow Fd$ to each morphism $f : c \rightarrow d$. It satisfies $F(1_c) = 1_{Fc}$ and $F(g \circ f) = Fg \circ Ff$ whenever $g \circ f$ is defined.

The extra requirement is not merely that objects be mapped to objects: the translation must respect how morphisms compose. A basic example is the direct image powerset endofunctor $\mathbb{P} : \mathbf{Set} \rightarrow \mathbf{Set}$. It sends a set X to its powerset $\mathbb{P}X$, and a function $f : X \rightarrow Y$ to the function $\mathbb{P}f : \mathbb{P}X \rightarrow \mathbb{P}Y$ defined by $(\mathbb{P}f)(U) = f[U]$.

Example 1. $\mathbb{P}(\{0, 1\}) = \{\emptyset, \{0\}, \{1\}, \{0, 1\}\}$.

The identity and composition laws follow from the corresponding laws for direct images.⁴

⁴ As an exercise, verify $\mathbb{P}(g \circ f) = \mathbb{P}g \circ \mathbb{P}f$.

REINFORCEMENT-LEARNING constructions can sometimes be made functorial. One first defines a category whose objects are Markov decision processes and whose morphisms are structure-preserving MDP homomorphisms,

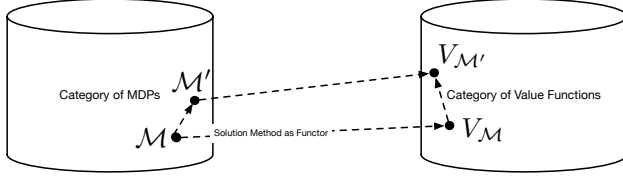


Figure 3: Reinforcement Learning (RL) methods can be modeled as functors.

and a second category of value-bearing objects with specified morphisms. An RL construction is functorial only if every MDP homomorphism induces a compatible map of value objects and the identity and composition laws hold. Merely assigning a value function V_M^π to each MDP M establishes the object part of a proposed functor, not functoriality by itself.

Functors can be viewed as a generalization of the notion of morphisms across algebraic structures, such as groups, vector spaces, and graphs. Functors do more than functions: they not only map objects to objects, but like graph homomorphisms, they need to also map each morphism in the domain category to a corresponding morphism in the co-domain category. Functors come in two varieties, as defined below.

Definition 8. A **covariant functor** $F : \mathcal{C} \rightarrow \mathcal{D}$ from category $\mathcal{C}$ to category $\mathcal{D}$, and defined as the following:

- An object FX (also written as $F(X)$) of the category $\mathcal{D}$ for each object X in category $\mathcal{C}$.
- An arrow $F(f) : FX \rightarrow FY$ in category $\mathcal{D}$ for every arrow $f : X \rightarrow Y$ in category $\mathcal{C}$.
- *Preservation of identity and composition:* $F(1_X) = 1_{FX}$ and $F(g \circ f) = Fg \circ Ff$ for composable arrows $f : X \rightarrow Y$ and $g : Y \rightarrow Z$.

Definition 9. A **contravariant functor** from $\mathcal{C}$ to $\mathcal{D}$ is a covariant functor $F : \mathcal{C}^{op} \rightarrow \mathcal{D}$. Equivalently, a morphism $f : X \rightarrow Y$ in $\mathcal{C}$ induces $Ff : FY \rightarrow FX$, with the identity and reversed-composition laws inherited from $\mathcal{C}^{op}$.

The *functoriality* axioms dictate how functors have to behave:

- For any composable pair f, g in category $\mathcal{C}$, $Fg \cdot Ff = F(g \cdot f)$.
- For each object c in $\mathcal{C}$, $F(1_c) = 1_{Fc}$.

We will use the notion $\mathcal{C}^{op}$ to denote the category $\mathcal{C}$ with the same objects, but with arrows reversed.

FUNCTOR CATEGORIES will play a major role in this book. For categories $\mathcal{C}$ and $\mathcal{D}$ satisfying the usual size conditions (for example, $\mathcal{C}$ small and $\mathcal{D}$ locally small), the functor category $[\mathcal{C}, \mathcal{D}]$ has functors as objects and natural transformations as morphisms. Presheaf categories are a particularly

important instance and are the setting of the Yoneda embedding developed in Chapter I.

Example 2. The presheaf category $[C^{op}, \mathbf{Set}]$, also written $\mathbf{Set}^{C^{op}}$, has functors $C^{op} \rightarrow \mathbf{Set}$ as objects and natural transformations as morphisms.

Example 3. The Yoneda embedding $\mathcal{Y} : C \rightarrow \mathbf{Set}^{C^{op}}$ sends c to the representable presheaf $\mathcal{Y}(c) = C(-, c)$. A morphism $u : c \rightarrow c'$ is sent to the natural transformation whose component maps $h : x \rightarrow c$ to $u \circ h : x \rightarrow c'$.

The construction characterizes an object through all morphisms into it. A token-context model may motivate a Yoneda-style analogy, but the ordinary Transformer architecture is not thereby a Yoneda embedding. Such an interpretation requires a declared token category, an enrichment, and a proof that the learned maps satisfy the relevant functorial laws [Bradley et al., 2021].

Example 4. The Yoneda embedding $\mathcal{Y} : C \rightarrow \mathbf{Set}^{C^{op}}$ is covariant in c . Its value at an object c is the representable presheaf $C(-, c)$, which is contravariant in its open argument. These are two distinct variances.

This example is highly instructive: verify both variance statements.⁵

⁵ Given $u : c \rightarrow c'$, postcomposition defines $C(-, c) \Rightarrow C(-, c')$; given $f : x \rightarrow x'$, a presheaf acts by precomposition $C(x', c) \rightarrow C(x, c)$.

NATURAL TRANSFORMATIONS are the morphisms of a functor category. Historically, Eilenberg and Mac Lane introduced categories and functors while studying natural transformations; the three notions were designed together rather than as a simple linear sequence of inventions [Mac Lane, 1971].

Definition 10. Given functors $F, G : C \rightarrow D$, a natural transformation $\alpha : F \Rightarrow G$ is a family of morphisms $\alpha_c : Fc \rightarrow Gc$, one for each object c of C , such that for every morphism $f : c \rightarrow c'$ the naturality equation

$$Gf \circ \alpha_c = \alpha_{c'} \circ Ff$$

holds. Thus the following square commutes [Riehl, 2017, Section 1.4].

$$\begin{array}{ccc} Fc & \xrightarrow{\alpha_c} & Gc \\ Ff \downarrow & & \downarrow Gf \\ Fc' & \xrightarrow{\alpha_{c'}} & Gc' \end{array}$$

NATURAL TRANSFORMATIONS AND FUNCTORS are two of the most important aspects of categorical AGI, and it takes some effort to understand their importance. Rather than delve completely in the technicalities of their definition, let us understand first their crucial role in transforming AI. Fundamentally, modern AI is based around *functions*. We think of Transformer

models as functions mapping sequences of tokens into other sequences of tokens, e.g., translating sentences in English into French or Japanese. Why do we need functors or natural transformations? The goal of this book and course is to explain their importance to AI and ML. At the end of this course, you will hopefully understand the importance of functors, and wonder why more attention is not being paid to them in AI research.

To begin with, let us ask the most basic question of all in machine learning: how do we define the problem of machine learning? A Transformer is essentially a nonlinear approximator of sequence-to-sequence functions. It works because it is possible to train a Transformer model using an enormous amount of data, specifically trillions of tokens. But, more abstractly, suppose we are given a finite sample drawn from some set (e.g., in training a Transformer, we are given samples of vectors $x \in \mathbb{R}^{n \times d}$). How do we define the problem of extrapolating (or interpolating) the data over some finite subset of token sequences to the entire vector space? This problem unfortunately has no canonical solution. Much of the literature in machine learning and statistics over many decades has explored a wide range of approaches.⁶

PAC (Probably Approximately Correct) learning formalizes one family of distributional learning problems [Valiant, 1984]. In the basic supervised setting, labeled examples $(x, f(x))$ are sampled independently from a fixed but unknown distribution $\mathbb{P}$ on an input space. With high probability over the sample, the learner must return a hypothesis g whose generalization error $\mathbb{P}_{x \sim \mathbb{P}}[g(x) \neq f(x)]$ is at most a prescribed ϵ , using resources controlled by the accuracy, confidence, and hypothesis-class parameters. Modern language-model training is not literally this elementary protocol. The point is narrower: a statistical learning problem becomes well posed only after its data-generating assumptions, hypothesis class, loss, and success criterion have been declared. Category theory can organize those declarations but cannot choose them canonically.

FUNCTOR EXTENSION supplies a different problem statement. Given $F : \mathcal{C} \rightarrow \mathcal{D}$ and $K : \mathcal{C} \rightarrow \mathcal{E}$, one may seek a functor on $\mathcal{E}$ related universally to F along K . When the required limits or colimits exist, the right and left Kan extensions provide canonical universal solutions to two dual approximation problems [Kan, 1958]. This does not make statistical learning canonical: the categories, morphisms, data semantics, and existence hypotheses still have to be chosen and justified. Kan extension is therefore a structural tool that can organize an extrapolation problem, not a replacement for its statistical assumptions.

⁶ Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning – with Applications in R*, volume 103 of *Springer Texts in Statistics*. Springer, New York, 2013. ISBN 978-1-4614-7137-0. DOI: 10.1007/978-1-4614-7138-7

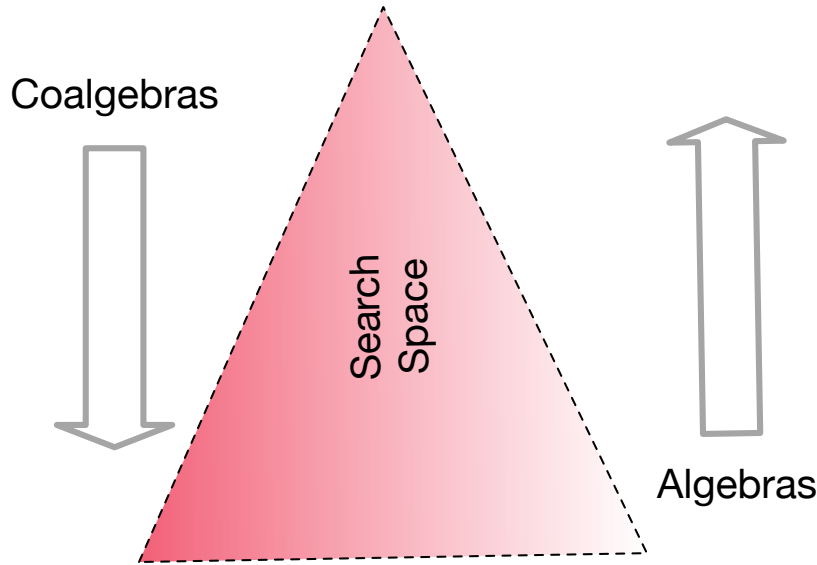


Figure 4: Coalgebras play a fundamental role in this book: they enable modeling infinite objects, search spaces, state, and enable reasoning about partial observability. Algebras are the dual to coalgebras: they compact and summarize information. Together, coalgebras and algebras enable modeling a wide range of AI and ML problems.

Generative AI with Universal Coalgebras

Coalgebras provide one of the book’s recurring languages for state-based and potentially unbounded behavior [Jacobs, 2016]. For an endofunctor $F : \mathcal{C} \rightarrow \mathcal{C}$, an F -coalgebra is a morphism $c : X \rightarrow FX$. The choice of F declares which observations and successor structures a state exposes. This covers streams, transition systems, automata, and several decision-process models. Coalgebras are not synonymous with every generative model; they become appropriate only after the relevant state type and behavioral functor have been specified.

Reasoning about Circularity in AI and ML

Many problems in AI and ML involve feedback or recursively specified behavior: common knowledge, infinite streams, transition systems, and cyclic causal models. Coalgebra handles many such systems without requiring their states themselves to be non-well-founded sets. Aczel’s anti-foundation axiom provides a separate set-theoretic semantics in which circular membership equations can be interpreted directly [Aczel, 1988]. The two perspectives are related, but they should not be identified.

- *Common knowledge*: the problem of modeling common knowledge occurs in a variety of real-world applications of AI. An autonomous car needs to model the behavior of other autonomous cars, as well as cars driven by humans. One fundamental problem here is modeling the rules of traffic. An autonomous car needs to obey local traffic laws, such as driving on the right side of the street in the United States (whereas in the United

Kingdom, the same car would need to drive on the left side of the street). But, it is not sufficient to just know which side of the street to drive on. The autonomous car needs to also model the knowledge of other drivers. The car needs to reason along the lines that it knows the other drivers also know which side of the street to drive on, and also it must know that the other drivers know that it knows the rules of the road, and so on. This problem of modeling common knowledge has long been a core problem in many areas related to AI, including philosophy, linguistics, and psychology.

- *Natural language utterances:* When humans communicate with each other, they are acutely aware of common knowledge and exploit them to keep their utterances as succinct as possible. Social networks involve posting a lot of short messages because the person posting does not need to repeat what he or she knows other people generally know about a particular world event. In everyday conversations between human beings, common knowledge is routinely used to simplify what needs to be said, otherwise humans would have to engage in interminably long conversations that always began at some initial point!
- *Modeling infinite data streams:* Many problems in AI and ML require modeling a potentially infinite data stream. A conversational chatbot that is trained on trillions of tokens comprised of hundreds of billions of web pages and other textual archives can be assumed to be operating in an environment where it is exposed to a never-ending data stream. Standard data structures in computer science, such as lists or trees, are defined recursively – a list is defined by its *head*, its first element, and its *tail*, or the elements that succeed the first element.

Three Representations for Reasoning about Circularity

In this chapter, we will explore three representations of the same concept, which together give deeper insight into how to model circular and recursive knowledge in AI and ML systems.

- *Universal coalgebras:* For a fixed endofunctor F , coalgebras are maps $X \rightarrow FX$. They describe one-step observations and successors, and hence support coinductive accounts of streams, automata, and dynamical systems. An infinite stream, for example, is exposed through its head and tail.
- *Non-well-founded sets:* Aczel's anti-foundation axiom supplies a semantics for guarded systems of circular set equations, such as graphical descriptions of self-membership. This is one possible semantics for circularity, not a prerequisite for ordinary coalgebra. The existence of a final F -coalgebra depends on F and the ambient category; it is not equivalent

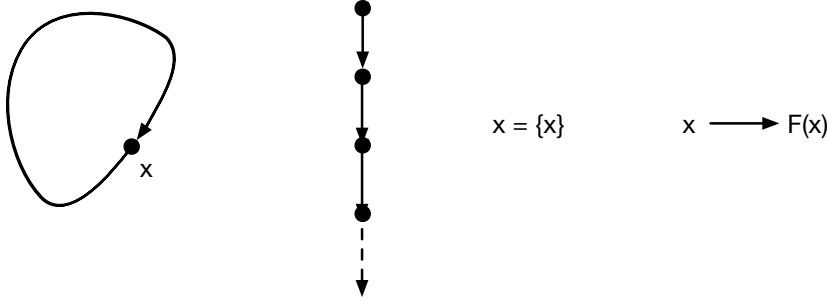


Figure 5: Three representations: accessible pointed graphs (AGPs), non-well-founded sets specified by systems of equations, and universal coalgebras.

to solving a cardinal equation such as $X \cong \mathbb{P}X$, which Cantor's theorem rules out in **Set**.

- *Accessible pointed graphs*: A visual way to illustrate a non-well-founded set, such as $x = \{x\}$ is through a directed graph, whose *decoration* precisely corresponds to the non-well-founded set. Peter Aczel, who pioneered the study of non-well-founded sets, used these type of graphs as visual ways to illustrate the ideas.

AI and ML Systems as Universal Coalgebras

For a fixed endofunctor F , an F -algebra has a structure map $FX \rightarrow X$, while an F -coalgebra has a structure map $X \rightarrow FX$. The reversal is useful: an algebra describes how a structured collection is assembled, whereas a coalgebra describes what can be observed or unfolded from a state. The category of F -coalgebras has coalgebra homomorphisms as its morphisms, which makes behavioral comparison a categorical question.

Search Spaces as Coalgebras

A fundamental principle that underlies most approaches in AI and ML is to envision the solution to many problems as *search in a state space*. The space is defined as a set of possible solutions, which may be states of a board game, possible generative models in a ML system, and so on. The transitions from one state to another define the search space. The goal is to find a solution or a state that satisfies some criterion, which is sometimes referred to as a *goal state*. We illustrate the fundamental principle that underlies our book – modeling AI and ML systems as universal coalgebras – by showing how to model state spaces as coalgebras.

The figure illustrates part of the state space for Tic Tac Toe. Let $\mathcal{S}$ be the set of legal game states and let $F = \mathbb{P}$ be the powerset endofunctor. The transition system is the F -coalgebra

$$\tau : \mathcal{S} \longrightarrow \mathbb{P}(\mathcal{S}),$$

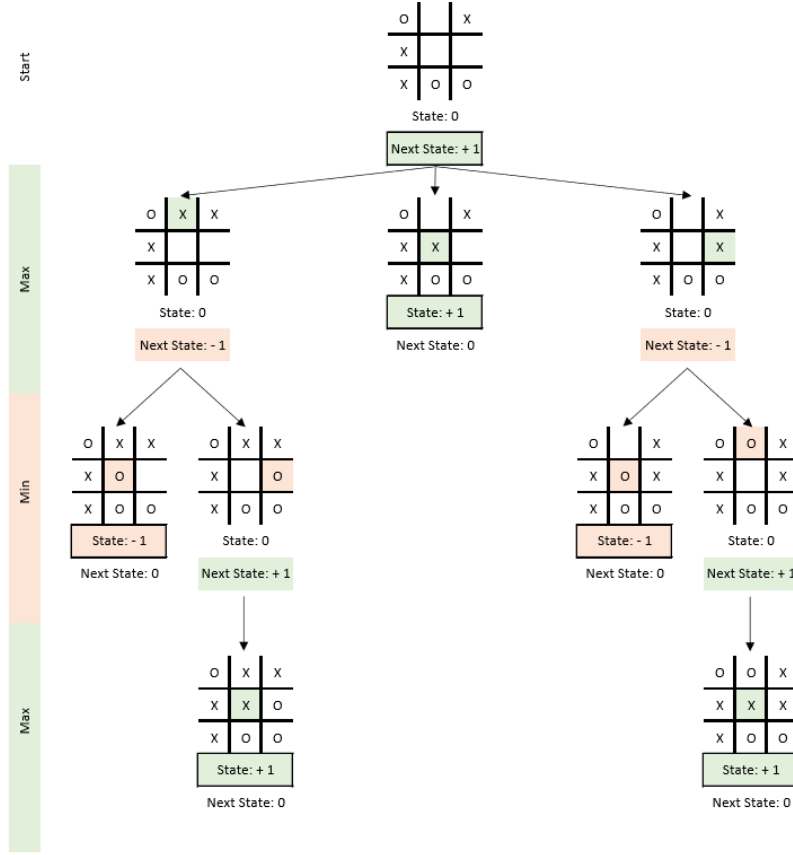


Figure 6: The search space for the game Tic Tac Toe can be modeled as a coalgebra.

where $\tau(s)$ is the set of legal one-step successors of s . The functor declares nondeterministic branching; the coalgebra map records the rules of this particular game.

We can elaborate this simple model in many interesting ways, as we will shortly see in the next section. For example, suppose the players cannot actually see the actual Tic Tac Toe board, but can only perceive some numerical value $f : \mathcal{S} \rightarrow \mathbb{N}$ that maps each state into a non-negative natural number. Such functions are common in the literature on heuristic search in AI, and are called *heuristic functions* as they embody some intuition about the desirability of a state. For example, one simple heuristic function might be to compute the number of ways in which a player can achieve a sequence of three elements along a row, column, or diagonal without being blocked by the opponent. Thus, in the beginning, the initial state has a score of 8 as there are eight possible ways to solve the Tic Tac Toe problem. In the final state where all the cells are filled, the score would be 0. We can consider more elaborate versions of this heuristic function that subtracts the number of ways the opponent might succeed from the player's own score. Regardless of which heuristic function is used, we can now model the revised search space

coalgebra for Tic Tac Toe as follows:

$$\tau_o : \mathcal{S} \longrightarrow \mathbb{N} \times \mathbb{P}(\mathcal{S}), \quad \tau_o(s) = (o(s), \tau(s)).$$

We can treat the heuristic value as a state observation. To record that observation, a state-associated cost, and a stochastic successor distribution, define $F(X) = \mathbb{N} \times \mathbb{R} \times \mathcal{D}(X)$, where $\mathcal{D}$ is a specified probability-distribution functor. A corresponding coalgebra is

$$\tau_s : \mathcal{S} \longrightarrow \mathbb{N} \times \mathbb{R} \times \mathcal{D}(\mathcal{S}).$$

Here $\mathcal{D}(\mathcal{S})$ is the set of distributions admitted by the chosen functor, for example finitely supported probability distributions. The first two components are state observations. A genuinely transition-dependent cost would require a different functor or an explicit edge-labeled presentation. This is why the behavioral functor must be declared rather than inferred from an informal description.

- *Unfolding*: Coalgebras can model finite or infinite behavior. The map $X \rightarrow FX$ exposes one observational step; repeated application may unfold an unbounded trajectory when the functor contains a recursive state component.
- *State*: The ability to model infinite streams in coalgebras gives them the ability to model state and *partial observability* of state in the real world. AI and ML systems fundamentally involve the notion of modeling states that represent points at which decisions are to be made. Both the self-driving car and the AI assistant require modeling states. Coalgebras provide an elegant and powerful way to capture not just state, but also *partial observability* of states in real world applications. As a simple example, a partially observable nondeterministic finite state machine can be defined by the coalgebra $X \rightarrow F(X)$
- *Category of coalgebras*: For a fixed endofunctor F , the F -coalgebras and their coalgebra homomorphisms form a category. A *final coalgebra*, when it exists, is a final object of this category and provides canonical behavioral semantics. Existence is a theorem about the chosen F and ambient category, not part of the definition of a coalgebra [Rutten, 2000].
- *Bisimulation*: Bisimulation compares states by their observable behavior and successors. Under suitable hypotheses it characterizes equality in final semantics. It applies to finite transition systems as well as to potentially infinite behavior.
- *Minimal representation*: When a final coalgebra exists, a coalgebra is often called observable or simple when its unique semantics map to the final coalgebra is monic; behaviorally equivalent states have already been identified. Other notions of coalgebraic minimality also impose reachability or

use quotient constructions. Thus monicity captures one precise aspect of minimality, not Occam’s razor in full generality.

- *Fixed points and final semantics:* Final coalgebras, when they exist, provide canonical behavioral semantics for F -coalgebras and support coinductive proofs. Bellman equations also involve fixed points, but a Bellman fixed point is not automatically a final coalgebra. Relating the two requires a separate construction specifying the behavioral functor, the value domain, and the comparison map.

Coalgebras model state-based systems, and several algebraic notions have coalgebraic duals or analogues: subcoalgebras describe subsystems, quotients can identify behaviorally equivalent states, and bisimulations provide a relation-based account of behavioral equivalence. The analogy is not exact for every endofunctor or ambient category. Under suitable hypotheses, final coalgebras support coinduction and can realize greatest-fixed-point semantics; neither their existence nor this interpretation is automatic [Rutten, 2000, Jacobs, 2016].

Clustering as a Functor

One of the book’s motifs is to formulate algorithms functorially. For clustering, this requires more than naming input and output objects. One must choose morphisms, such as non-expansive maps between finite metric spaces, and a category of partitions whose morphisms respect refinement. A clustering functor must carry every admissible metric-space morphism to a compatible partition morphism. Finite metric spaces need not be embedded in $\mathbb{R}^n$; they are finite sets equipped with a metric. The precise categorical formulation is developed by Carlsson and Mémoli [2013].

One can impose three criteria on a clustering algorithm, which seem entirely natural, and yet, no standard clustering algorithm satisfies all these conditions.⁷

- *Scale invariance:* If the distance metric d is increased or decreased by $c \cdot d$, where c is a scalar real number, the output clustering should not change. If the points in each cluster became closer together or further apart proportionally, the clustering should remain the same.
- *Completeness:* For any given partition of the space X , there should exist some distance function d such that the clustering algorithm when given that distance function should return the desired partition.
- *Monotonicity:* If the distance between points within each cluster were decreased, and the distances between points in different clusters were increased, the clustering should not change either.

⁷ Jon Kleinberg. An impossibility theorem for clustering. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15. MIT Press, 2002

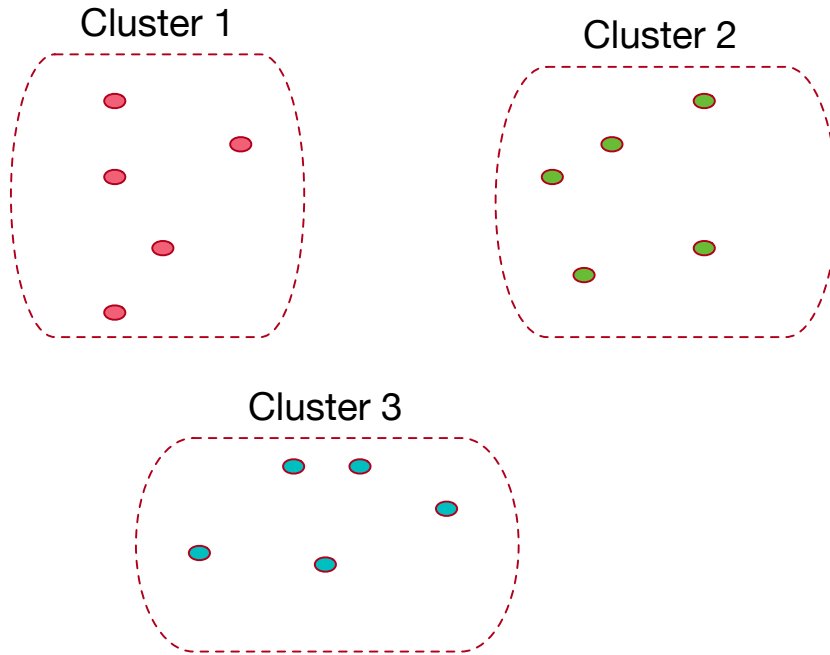


Figure 7: Clustering constructs partitions from finite metric data. Functoriality concerns compatibility with declared morphisms; scale invariance is a separate axiom in Kleinberg’s formulation.

Kleinberg proves that no clustering function on finite metric spaces satisfies all three axioms simultaneously [Kleinberg, 2002]. This is an impossibility theorem for that one-scale formulation. Persistent and hierarchical clustering change the output type by retaining information across scales; within that reformulated problem, nontrivial functorial classification results become possible [Carlsson and Mémoli, 2013]. They do not refute Kleinberg’s theorem; they alter its premises.

Summary

This chapter introduced functors, natural transformations, and coalgebras. The reader should now be able to separate an objectwise assignment from a genuine functor by checking preservation of identities and composition.⁸

A great source for understanding universal coalgebras is Rutten’s classic chapter.⁹ Using simple and elegant diagrams, he characterizes a wide spectrum of dynamical systems in terms of their fundamental coalgebraic structure.

UMAP (Uniform Manifold Approximation and Projection) is a widely used nonlinear dimension-reduction method. Its construction draws on fuzzy simplicial sets and mappings between metric and combinatorial representations, offering a concrete example of structure-preserving design. The software documentation provides an accessible computational introduction.¹⁰

⁸ Choose explicit categories of MDPs and value-bearing objects. Under what assumptions does an RL construction become a functor, and what would a natural transformation between two such functors require?

⁹ J.J.M.M. Rutten. Universal coalgebra: A theory of systems. *Theoretical Computer Science*, 249(1):3–80, 2000. ISSN 0304-3975. DOI: 10.1016/S0304-3975(00)00056-6. URL [https://doi.org/10.1016/S0304-3975\(00\)00056-6](https://doi.org/10.1016/S0304-3975(00)00056-6). Modern Algebra

¹⁰ https://umap-learn.readthedocs.io/en/latest/how_umap_works.html

Chapter I now asks what can be learned from an entire functorial profile of relationships. Natural transformations become the morphisms whose structure makes the Yoneda Lemma possible.

Suggested Reading

[Mac Lane \[1971\]](#) and [Riehl \[2017\]](#) give the standard treatment of functors and natural transformations. For learning systems, [Fong et al. \[2019\]](#) organizes backpropagation functorially. The coalgebraic perspective on dynamical systems is developed by [Rutten \[2000\]](#) and [Jacobs \[2016\]](#).

Representable Functors and the Yoneda Lemma

This chapter develops representable functors and the Yoneda Lemma, one of the central results of elementary category theory.¹¹ Informally, an object is determined up to isomorphism by the complete, functorial profile of morphisms into it (or, dually, out of it). The words *complete* and *functorial* matter: Yoneda is not the claim that any collection of pairwise similarities uniquely identifies an object.

Later chapters use representables as a design language for relational representations, causal models, and enriched attention. Those are proposed applications of Yoneda’s pattern, not consequences of the lemma alone. The “Universal Causality” construction is one such application [Mahadevan, 2023]; its causal interpretation requires additional semantic assumptions developed in Section IV.

¹¹ A detailed discussion appears in Chapter 2 of Riehl’s textbook.

Natural Transformations and Universal Arrows

Given any two functors $F : C \rightarrow D$ and $G : C \rightarrow D$ between the same pair of categories, we can define a mapping between F and G that is referred to as a natural transformation. These are defined through a collection of mappings, one for each object c of C , thereby defining a morphism in D for each object in C .

Definition 11. Given categories C and D , and functors $F, G : C \rightarrow D$, a **natural transformation** $\alpha : F \Rightarrow G$ is defined by the following data:

- an arrow $\alpha_c : Fc \rightarrow Gc$ in D for each object $c \in C$, which together define the components of the natural transformation.
- For each morphism $f : c \rightarrow c'$, the following commutative diagram holds true:

$$\begin{array}{ccc} Fc & \xrightarrow{\alpha_c} & Gc \\ Ff \downarrow & & \downarrow Gf \\ Fc' & \xrightarrow{\alpha_{c'}} & Gc' \end{array}$$

A **natural isomorphism** is a natural transformation $\alpha : F \Rightarrow G$ in which every component α_c is an isomorphism.

A fundamental universal construction in category theory, called the *universal arrow* lies at the heart of many useful results, principally the Yoneda lemma that shows how object identity itself emerges from the structure of morphisms that lead into (or out of) it.

Definition 12. Given a functor $S : D \rightarrow C$ between two categories, and an object c of category C , a **universal arrow** from c to S is a pair $\langle r, u \rangle$, where r is an object of D and $u : c \rightarrow Sr$ is an arrow of C , such that the following universal property holds true:

- For every pair $\langle d, f \rangle$ with d an object of D and $f : c \rightarrow Sd$ an arrow of C , there is a unique arrow $f' : r \rightarrow d$ of D with $Sf' \circ u = f$.

Once again, it is important to pause here, and try to understand what this definition is stating.¹²

¹² Draw the commutative diagram for this definition!

Definition 13. If D is a category and $H : D \rightarrow \mathbf{Set}$ is a set-valued functor, a **universal element** associated with the functor H is a pair $\langle r, e \rangle$ consisting of an object $r \in D$ and an element $e \in Hr$ such that for every pair $\langle d, x \rangle$ with $x \in Hd$, there is a unique arrow $f : r \rightarrow d$ of D such that $(Hf)e = x$.

A UNIVERSAL PROPERTY characterizes an object as initial or terminal in an appropriate category of candidates. It determines the object up to a unique isomorphism compatible with the universal data. The phrase “universal” therefore refers to a quantified factorization property, not merely to a construction that is broad or frequently useful.

Theorem 1. Given a functor $S : D \rightarrow C$, a universal arrow $\langle r, u : c \rightarrow Sr \rangle$ induces a bijection

$$D(r, d) \cong C(c, Sd)$$

for every $d \in D$, and these bijections are natural in d .

Proof. Define $\Phi_d : D(r, d) \rightarrow C(c, Sd)$ by $\Phi_d(h) = Sh \circ u$. The defining factorization property of the universal arrow says exactly that every $f : c \rightarrow Sd$ has a unique preimage $h : r \rightarrow d$, so Φ_d is bijective. For $k : d \rightarrow d'$, functoriality gives $\Phi_{d'}(k \circ h) = S(k \circ h) \circ u = Sk \circ \Phi_d(h)$, which is the required naturality square. This is the standard representation theorem for universal arrows [Mac Lane, 1971, Section III.1]. $\square$

The Yoneda Lemma makes the role of the identity element precise: a natural transformation from a representable functor is determined by its value on the identity morphism of the representing object.

Yoneda Lemma

$$\begin{array}{ccc}
 C(c, x) & \xrightarrow{\alpha_x} & Fx \\
 \downarrow C(c, u) & & \downarrow Fu \\
 C(c, y) & \xrightarrow{\alpha_y} & Fy
 \end{array}$$

For every morphism $u : x \rightarrow y$, this square is the naturality condition for a transformation $\alpha : C(c, -) \Rightarrow F$.

For a locally small category $\mathcal{C}$, the assignment $d \mapsto \mathcal{C}(-, d)$ is a fully faithful functor

$$\mathcal{Y} : \mathcal{C} \longrightarrow \mathbf{Set}^{\mathcal{C}^{op}}.$$

Thus the representable presheaf $\mathcal{C}(-, d)$ determines d up to isomorphism. If $\mathcal{C}$ is small, the presheaf category $\mathbf{Set}^{\mathcal{C}^{op}}$ is a Grothendieck topos. The topos belongs to the ambient presheaf construction; Yoneda does not assert that $\mathcal{C}$ itself is a topos.

$$\begin{array}{ccc}
 D(r, r) & \xrightarrow{\phi_r} & C(c, Sr) \\
 \downarrow D(r, f') & & \downarrow C(c, Sf') \\
 D(r, d) & \xrightarrow{\phi_d} & C(c, Sd)
 \end{array}$$

The square expresses naturality of the bijections in d . A universal arrow from c to S is equivalently a representation of the functor $C(c, S-) : D \rightarrow \mathbf{Set}$ by the object r : there is a natural isomorphism $D(r, -) \cong C(c, S-)$. Local smallness ensures that each hom-set is a set; smallness is the stronger condition that the objects and morphisms of the entire category form sets.

Lemma 1. Yoneda lemma: *For any functor $F : \mathcal{C} \rightarrow \mathbf{Set}$, whose domain category $\mathcal{C}$ is “locally small” (meaning that the collection of morphisms between each pair of objects forms a set), any object c in $\mathcal{C}$, there is a bijection*

$$\mathrm{Nat}(C(c, -), F) \simeq Fc$$

that defines a natural transformation $\alpha : C(c, -) \Rightarrow F$ to the element $\alpha_c(1_c) \in Fc$. This correspondence is natural in both c and F .

Proof. Send $\alpha : C(c, -) \Rightarrow F$ to $\alpha_c(1_c)$. Conversely, for $x \in Fc$, define $\alpha_d^x : C(c, d) \rightarrow Fd$ by $\alpha_d^x(f) = Ff(x)$. Functoriality of F proves naturality of α^x . The naturality square for any $f : c \rightarrow d$, applied to 1_c , gives $\alpha_d^x(f) = Ff(\alpha_c(1_c))$; hence the two constructions are inverse. Naturality in c and F follows by substitution. See [Riehl \[2017, Theorem 2.2.4\]](#) or [Mac Lane \[1971, Section III.2\]](#). $\square$

There is a dual presheaf form: for $P : \mathcal{C}^{op} \rightarrow \mathbf{Set}$, $\mathrm{Nat}(C(-, c), P) \cong P(c)$. Representability is the standard name for a functor naturally isomorphic to one of these hom-functors.

Definition 14. A functor $P : C^{op} \rightarrow \mathbf{Set}$ is **representable** when there is an object $c \in C$ and a natural isomorphism $P \cong C(-, c)$. Dually, a covariant functor $F : C \rightarrow \mathbf{Set}$ is **representable** when $F \cong C(c, -)$. A **representing object**, when it exists, is unique up to unique isomorphism compatible with the chosen representation.

Yoneda Intuition via Visualizing Self-Attention in BERT

A SLOGAN that will guide this chapter is the following: *an object is determined by all of its relationships*. The Yoneda Lemma makes this precise by identifying an object $c \in C$ with the functor of morphisms out of (or into) it, such as $C(c, -)$ or $C(-, c)$, and by stating that natural transformations out of a representable functor are equivalent to elements of the target functor.

SELF-ATTENTION offers a useful but limited analogy. A token representation is context-sensitive because its update depends on learned relations to other token positions. This resembles the relational slogan behind Yoneda, but an attention vector is not generally a representable functor and the attention equations do not by themselves satisfy a Yoneda universal property.

ATTENTION AS A CANDIDATE ENRICHED PROFILE. Let A denote a token position in a sentence. In a toy causal “sequence category” (as in the Week 2 micro-lab), the representable functor $h_A(-) = \text{Hom}(A, -)$ is the set of arrows from A to all other token positions. In a Transformer, self-attention constructs a weighted profile:

$$\alpha_{A \rightarrow j} = \text{softmax}_j \left(\frac{\langle q_A, k_j \rangle}{\sqrt{d}} \right), \quad y_A = \sum_j \alpha_{A \rightarrow j} v_j,$$

where q_A is the query vector for token A , k_j is the key vector for token j , and v_j is the value. Thus y_A aggregates values along learned relations from A . Interpreting this profile as enriched categorical structure requires an enrichment base, composition law, and unit; these are additional modeling commitments rather than properties supplied by softmax alone.

A DISAMBIGUATION DEMO: “NEW ENGLAND”. Consider the two sentences:

- (1) *The New England Patriots will win Super Bowl 2026.*
- (2) *New England winters can be really cold and long.*

The phrase *New England* is ambiguous: in (1) it refers primarily to a sports team brand/identity, while in (2) it refers to a geographic/climatic region. A pretrained model (e.g. BERT) resolves this ambiguity by producing different contextual encodings for the tokens *new* and *england*, which can be observed

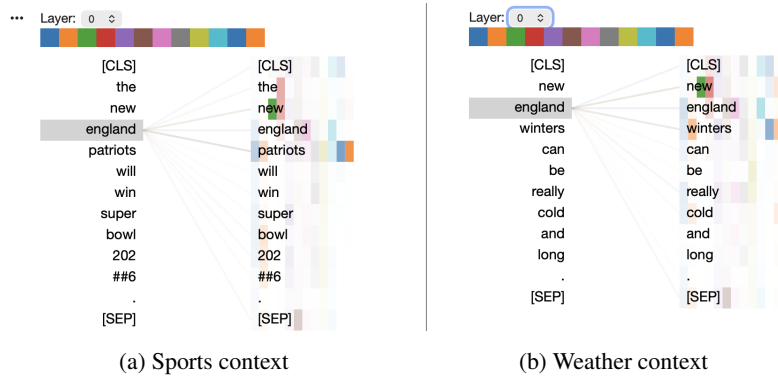


Figure 8: **Attention as a Yoneda-style relational profile.** Clicking the token *england* reveals its head-wise attention distribution $\alpha_{A \rightarrow -}$ at a fixed layer. The attention profile—and therefore the contextual embedding—changes with context, disambiguating the phrase *New England*. Source code: adapted from the BertViz example in the [official Chapter 3 companion notebook](#) to *Natural Language Processing with Transformers*; full bibliographic citations appear in the preceding text.

directly by visualizing attention weights. In (1), the attention profiles for *england* place mass on tokens such as *patriots*, *super*, *bowl*; in (2) they place mass on *winters*, *cold*, *long*. In other words, the “relationship profile” of the token changes with context, and the token’s representation changes accordingly.

VISUALIZATION. Figures 8a and 8b show BertViz attention-head visualizations for the two contexts. They were produced by adapting the BertViz workflow in the official Chapter 3 companion notebook for *Natural Language Processing with Transformers* [Tunstall et al., 2022b,a] to the two sentences used here.

CONNECTION BACK TO YONEDA. The Yoneda Lemma states that an object c is determined by the functor $C(c, -)$ (or $C(-, c)$). Self-attention can be viewed heuristically as learning a soft relational analogue: a token representation is updated from a weighted profile of its relationships to other tokens. This analogy motivates categorical models; it does not identify ordinary attention with the Yoneda embedding. In subsequent chapters, we will formalize how diagrammatic constraints (e.g. commutative squares) can be used to regularize such relationship profiles and how sheaf/topos structure can be used to glue local views into globally coherent commitments.

Summary

This chapter established the Yoneda Lemma and the fully faithful Yoneda embedding. Its precise content concerns natural transformations from representable functors, not arbitrary relational data. Digital profiles, social networks, attention maps, and causal influence diagrams may all invite Yoneda-style models, but each application must specify its category and prove that the proposed representation respects the relevant morphisms.

An alternate form of the Yoneda Lemma makes it somewhat easier to appreciate:

$$\mathrm{Nat}(\mathcal{C}(-, c), \mathcal{C}(-, d)) \simeq \mathcal{C}(c, d)$$

The isomorphism reconstructs morphisms $c \rightarrow d$ from natural transformations between representables. It acquires a causal reading only when the morphisms of $\mathcal{C}$ have independently been given causal semantics; Yoneda alone does not identify statistical or causal influence.

Chapter I turns this relational viewpoint into a construction language: diagrams organize many objects at once, while limits and colimits select universal ways of comparing or assembling them.

Suggested Reading

The Yoneda lemma and representability are treated carefully by [Mac Lane \[1971\]](#) and [Riehl \[2017\]](#). For an application-facing account, [Spivak \[2014\]](#) develops the relational viewpoint through scientific examples. [Bradley et al. \[2021\]](#) applies enriched categorical ideas to syntax and semantics.

Diagrams and Universal Constructions

This chapter introduces diagrams, limits, and colimits. A diagram records a family of objects and morphisms with a specified shape; a limit or colimit, when it exists, summarizes that diagram through a universal mapping property. These constructions provide a language for assembling and comparing model components. Their existence in an ambient category such as **Set** or a presheaf topos does not imply that the result is realizable by a chosen neural architecture. That closure question must be established separately.

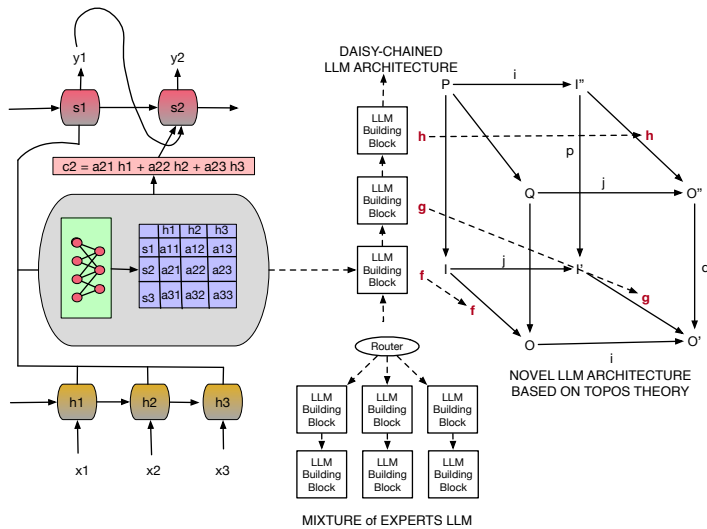


Figure 9: Limits and colimits provide abstract construction patterns for model architectures. Realizability by a particular neural family is an additional question.

A key distinguishing feature of category theory is the use of *diagrammatic reasoning*. However, diagrams are also viewed more abstractly as functors mapping from some indexing category to the actual category. Diagrams are useful in understanding universal constructions, such as limits and colimits of diagrams.¹³ These concepts are among the most difficult to understand initially, and it will require much concentration to absorb their implications. As with all beautiful abstractions in pure mathematics, a deeper understanding of these ideas will reward the reader with a much deeper appreciation for their application to AGI.

DIAGRAMS are functors in disguise from some *indexing* category (usually

¹³ Read Chapter 3 of Riehl's textbook for a great introduction to these concepts.

finite) to an actual concrete category of interest. Let us consider as a simple example the concept of a *Cartesian* product of sets, which is usually defined as

$$A \times B = \{(a, b) | a \in A, b \in B\}$$

Given two objects c and d , their categorical product is characterized without inspecting internal elements. It is an object $c \times d$ equipped with projections to c and d such that every compatible pair of maps from an object x factors uniquely through $c \times d$. In **Set**, this universal property recovers the ordinary set of ordered pairs.

Definition 15. A diagram $F : \mathcal{J} \rightarrow \mathcal{C}$ is a functor from an indexing category $\mathcal{J}$ into a category $\mathcal{C}$. The indexing category may be finite, but need not be.

For example, $\mathcal{J} = \bullet \rightarrow \bullet \leftarrow \bullet$ is the shape of a cospan; its limit, when it exists, is a pullback. A category is complete (respectively cocomplete) when every small diagram has a limit (respectively colimit). Individual diagrams may have limits even when their ambient category is not complete.

Before we formally the concept of limit and colimits, we consider some examples. These notions generalize the more familiar notions of Cartesian products and disjoint unions in the category of **Sets**, the notion of meets and joins in the category **Preord** of preorders, as well as the least upper bounds and greatest lower bounds in lattices, and many other concrete examples from mathematics.

Example 5. If we consider a small “discrete” category $\mathcal{D}$ whose only morphisms are identity arrows, then the colimit of a functor $\mathcal{F} : \mathcal{D} \rightarrow \mathcal{C}$ is the categorical coproduct of $\mathcal{F}(D)$ for D , an object of category $\mathcal{D}$, is denoted as

$$\text{Colimit}_{\mathcal{D}} \mathcal{F} = \bigsqcup_D \mathcal{F}(D)$$

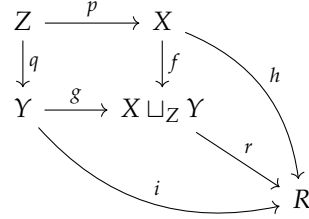
In the special case when the category $\mathcal{C}$ is the category **Sets**, then the colimit of this functor is simply the disjoint union of all the sets $\mathcal{F}(D)$ that are mapped from objects $D \in \mathcal{D}$.

Example 6. Dual to the notion of colimit of a functor is the notion of limit. Once again, if we consider a small “discrete” category $\mathcal{D}$ whose only morphisms are identity arrows, then the limit of a functor $\mathcal{F} : \mathcal{D} \rightarrow \mathcal{C}$ is the categorical product of $\mathcal{F}(D)$ for D , an object of category $\mathcal{D}$, is denoted as

$$\text{limit}_{\mathcal{D}} \mathcal{F} = \prod_D \mathcal{F}(D)$$

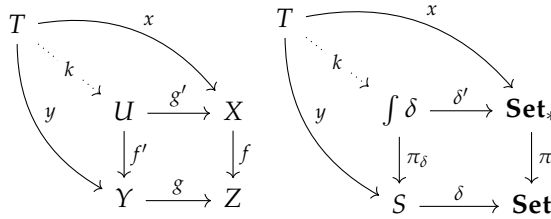
In the special case when the category $\mathcal{C}$ is the category **Sets**, then the limit of this functor is simply the Cartesian product of all the sets $\mathcal{F}(D)$ that are mapped from objects $D \in \mathcal{D}$.

Category theory relies extensively on universal constructions. Pullbacks are particular limits, while pushouts are their colimit duals and often formalize gluing. A pushout should not be confused with the pushforward of an object or measure along a map; the two notions may interact in applications but have different definitions.



The diagram displays the universal pattern of a pushout of the span $X \leftarrow Z \rightarrow Y$. Its pushout, conventionally written $X \sqcup_Z Y$, receives maps from X and Y that agree on Z ; every other agreeing pair $X \rightarrow R$ and $Y \rightarrow R$ factors uniquely through $X \sqcup_Z Y$. A coproduct is the special case in which the common source is initial.

The next figure illustrates a pullback. Given a cospan $X \xrightarrow{f} Z \xleftarrow{g} Y$, a pullback consists of an object U and maps $g' : U \rightarrow X$ and $f' : U \rightarrow Y$ satisfying $f \circ g' = g \circ f'$. For every other cone $T \xrightarrow{x} X$, $T \xrightarrow{y} Y$ satisfying $f \circ x = g \circ y$, there is a unique $k : T \rightarrow U$ with $g' \circ k = x$ and $f' \circ k = y$. Reversing all arrows gives the definition of a pushout.



PULLBACKS AND PUSHOUTS are dual universal constructions. A pullback is terminal among cones over a cospan; a pushout is initial among cocones under a span. Yoneda explains why these complete mapping properties determine the resulting objects up to unique isomorphism.

For a covariant functor $\delta : S \rightarrow \mathbf{Set}$, the category of elements is the strict pullback in $\mathbf{Cat}$ of $\delta : S \rightarrow \mathbf{Set}$ along the forgetful functor $\pi : \mathbf{Set}_* \rightarrow \mathbf{Set}$, if equality of the underlying sets is used in the pullback. If the square is required to commute only up to a specified isomorphism, the corresponding invariant construction is a pseudopullback. Here $\mathbf{Set}_*$ is the category of pointed sets $(X, x \in X)$ [Riehl, 2017, Section 2.4].

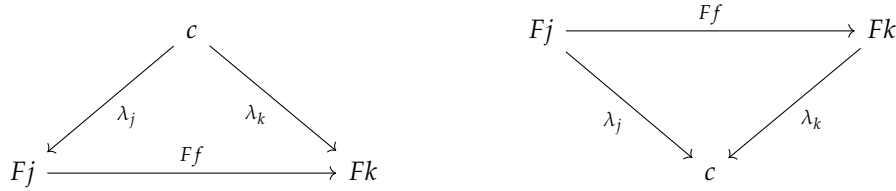
We can now proceed to define limits and colimits more generally. We define a *diagram* F of *shape* J in a category C formally as a functor $F : J \rightarrow C$. We want to define the somewhat abstract concepts of *limits* and *colimits*, which will play a central role in this book in identifying properties of AI and

Figure 10: **(Left)** Universal Property of pullback mappings. **(Right)** The Grothendieck category of elements $\int \delta$ of any set-valued functor $\delta : S \rightarrow \mathbf{Set}$ can be described as a pullback in the diagram of categories. Here, $\mathbf{Set}_*$ is the category of pointed sets $(X, x \in X)$, and π is the “forgetful” functor that sends a pointed set $(X, x \in X)$ into the underlying set X .

ML techniques. A convenient way to introduce these concepts is through the use of *universal cones* that are *over* and *under* a diagram.

For any object $c \in C$ and any category J , the *constant functor* $c : J \rightarrow C$ maps every object j of J to c and every morphism f in J to the identity morphisms 1_c . We can define a constant functor embedding as the collection of constant functors $\Delta : C \rightarrow C^J$ that send each object c in C to the constant functor at c and each morphism $f : c \rightarrow c'$ to the constant natural transformation, that is, the natural transformation whose every component is defined to be the morphism f .

Definition 16. A **cone over** a diagram $F : J \rightarrow C$ with apex $c \in C$ is a natural transformation $\lambda : \Delta c \Rightarrow F$, where Δc is the constant functor at c . Its components $\lambda_j : c \rightarrow Fj$ are its legs. Dually, a **cocone under** F with nadir c is a natural transformation $\lambda : F \Rightarrow \Delta c$ with legs $\lambda_j : Fj \rightarrow c$.



Cones under a diagram are referred to usually as *cocones*. Using the concept of cones and cocones, we can now formally define the concept of limits and colimits more precisely.

Definition 17. For any diagram $F : J \rightarrow C$, there is a functor

$$\text{Cone}(-, F) : C^{\text{op}} \rightarrow \mathbf{Set}$$

which sends $c \in C$ to the set of cones over F with apex c . A **limit** of F is an object $\lim F \in C$ together with a universal cone $\lambda : \Delta(\lim F) \Rightarrow F$, defining the natural isomorphism

$$C(-, \lim F) \simeq \text{Cone}(-, F)$$

Dually, for colimits, we define the cocone functor

$$\text{Cocone}(F, -) : C \rightarrow \mathbf{Set}$$

that maps $c \in C$ to the set of cocones under F with nadir c . A **colimit** of F is a representing object for this functor: an object $\text{colim } F$ together with a universal cocone $\lambda : F \Rightarrow \Delta(\text{colim } F)$ defining the natural isomorphism

$$C(\text{colim } F, -) \simeq \text{Cocone}(F, -)$$

Limit and colimits of diagrams over arbitrary categories can often be reduced to the case of their corresponding diagram properties over sets. One

important stepping stone is to understand how functors interact with limits and colimits.

Definition 18. For any class of diagrams $K : J \rightarrow C$, a functor $F : C \rightarrow D$

- **preserves limits** if for any diagram $K : J \rightarrow C$ and limit cone over K , the image of the cone defines a limit cone over the composite diagram $FK : J \rightarrow D$.
- **reflects limits** if any cone over a diagram $K : J \rightarrow C$ whose image under F is a limit cone over $FK : J \rightarrow D$ is itself a limit cone over K ;
- **creates limits of the specified shape** if, whenever FK has a limit in D , the diagram K has a limit in C , and a cone over K is limiting if and only if its image under F is limiting. The stronger notion of strict creation additionally requires a unique lift of each chosen limiting cone. Later claims will distinguish these conventions explicitly [Riehl, 2017, Section 3.4].

To interpret these abstract definitions, it helps to concretize them in terms of a specific universal construction, like the pullback defined above $c' \rightarrow c \leftarrow c''$ in C . Specifically, for pullbacks:

- A functor F **preserves pullbacks** if whenever p is the pullback of $c' \rightarrow c \leftarrow c''$ in C , it follows that Fp is the pullback of $Fc' \rightarrow Fc \leftarrow Fc''$ in D .
- A functor F **reflects pullbacks** if p is the pullback of $c' \rightarrow c \leftarrow c''$ in C whenever Fp is the pullback of $Fc' \rightarrow Fc \leftarrow Fc''$ in D .
- A functor F **creates pullbacks** when a chosen pullback cone of $Fc' \rightarrow Fc \leftarrow Fc''$ that lifts along F has a unique lifted cone over $c' \rightarrow c \leftarrow c''$, and the lifted cone is a pullback.

Universality of Diagrams

In the category **Sets**, we know that every object (i.e., a set) X can be expressed as a coproduct (i.e., disjoint union) of its elements $X \simeq \sqcup_{x \in X} \{x\}$, where $x \in X$. Note that we can view each element $x \in X$ as a morphism $x : \{*\} \rightarrow X$ from the one-point set to X . The categorical generalization of this result is called the *density theorem* in the theory of sheaves. First, we define the key concept of a *comma category*.

Definition 19. Let $F : \mathcal{D} \rightarrow \mathcal{C}$ be a functor and fix an object $C \in \mathcal{C}$. The comma category $(F \downarrow C)$ has objects (D, f) with $D \in \mathcal{D}$ and $f : F(D) \rightarrow C$. A morphism $(D, f) \rightarrow (D', f')$ is a morphism $g : D \rightarrow D'$ satisfying $f' \circ Fg = f$. It is depicted by the following commutative triangle:

$$\begin{array}{ccc} & F(D) & \\ & \swarrow \scriptstyle F(g) \quad \searrow \scriptstyle f & \\ F(D') & \xrightarrow{\scriptstyle f'} & C \end{array}$$

We first introduce the concept of a *dense* functor:

Definition 20. Let $\mathcal{D}$ be small, $\mathcal{C}$ be locally small, and $F : \mathcal{D} \rightarrow \mathcal{C}$ be a functor. The functor F is **dense** if every object C of $\mathcal{C}$ is canonically the colimit of the diagram of F -objects mapping to C ; equivalently, the nerve $\mathcal{C}(F-, -) : \mathcal{C} \rightarrow \mathbf{Set}^{\mathcal{D}^{op}}$ is fully faithful. In colimit form, the canonical cocone

$$\psi_F^C : F \circ U \rightarrow \Delta_C, \quad (\psi_F^C)_{(\mathcal{D}, f)} = f$$

is universal and induces an isomorphism $\text{colim}_{(F \downarrow \mathcal{C})} F \circ U \cong C$. Here, $U : (F \downarrow \mathcal{C}) \rightarrow \mathcal{D}$ is the projection $U(D, f) = D$.

A fundamental consequence of the category of elements is that every object in the functor category of presheaves, namely contravariant functors from a category into the category of sets, is the colimit of a diagram of representable objects, via the Yoneda lemma. Notice this is a generalized form of the density notion from the category **Sets**.

Theorem 2. Density theorem for presheaves: If $\mathcal{C}$ is small, then every presheaf $P : \mathcal{C}^{op} \rightarrow \mathbf{Set}$ is canonically a colimit of representables. More precisely,

$$P \cong \text{colim}_{(c, x) \in \int P} \mathcal{C}(-, c),$$

where $\int P$ is the category of elements of P .

Here $\int P$ has objects (c, x) with $x \in P(c)$; a morphism $(c, x) \rightarrow (d, y)$ is a morphism $f : c \rightarrow d$ satisfying $P(f)(y) = x$. This convention makes the projection $\int P \rightarrow \mathcal{C}$ covariant.

Proof. For each $(c, x) \in \int P$, the presheaf form of the Yoneda lemma identifies $x \in P(c)$ with a natural transformation $\mathcal{C}(-, c) \Rightarrow P$. These transformations form a cocone. At an object $d \in \mathcal{C}$, its induced map sends the class represented by $f : d \rightarrow c$ and $x \in P(c)$ to $P(f)(x) \in P(d)$. It is surjective because $y \in P(d)$ is represented by $(d, y, 1_d)$, and the morphisms of $\int P$ impose exactly the identifications required for injectivity. Colimits in a presheaf category are computed pointwise, so the resulting natural transformation is an isomorphism. This is the standard density theorem [Riehl, 2017, Theorem 6.5.8]. $\square$

Universal Arrows and Elements

A special case of the universal arrow property is that of universal element, which as we will see below plays an important role in the GAIA architecture in defining a suitably augmented category of elements, based on a construction introduced by Grothendieck.

Definition 21. If D is a category and $H : D \rightarrow \mathbf{Set}$ is a set-valued functor, a **universal element** associated with the functor H is a pair $\langle r, e \rangle$ consisting of

an object $r \in D$ and an element $e \in Hr$ such that for every pair $\langle d, x \rangle$ with $x \in Hd$, there is a unique arrow $f : r \rightarrow d$ of D such that $(Hf)e = x$.

Example 7. Let E be an equivalence relation on a set S , and define $H_E : \mathbf{Set} \rightarrow \mathbf{Set}$ by

$$H_E(X) = \{f : S \rightarrow X \mid s E s' \Rightarrow f(s) = f(s')\}, \quad H_E(k)(f) = k \circ f.$$

Let $p : S \rightarrow S/E$ be the quotient map. Then $p \in H_E(S/E)$, and every $f \in H_E(X)$ factors uniquely as $f = \tilde{f} \circ p$ for a function $\tilde{f} : S/E \rightarrow X$. Thus $\langle S/E, p \rangle$ is a universal element of the explicitly defined functor H_E ; equivalently, $H_E \cong \mathbf{Set}(S/E, -)$.

The Category of Elements

We turn next to define the category of elements, based on a construction by Grothendieck, and illustrate how it can serve as the basis for inference at each layer of the GAIA architecture.

Definition 22. Given a set-valued covariant functor $\delta : \mathcal{C} \rightarrow \mathbf{Set}$, its **category of elements** $\int \delta$ is equipped with a projection $\pi_\delta : \int \delta \rightarrow \mathcal{C}$. Its objects and arrows are defined as follows:

- $Ob(\int \delta) = \{(s, x) \mid s \in Ob(\mathcal{C}), x \in \delta(s)\}$.
- $\mathbf{Hom}_{\int \delta}((s, x), (s', x')) = \{f : s \rightarrow s' \mid \delta f(x) = x'\}$

Summary

This chapter introduced diagrams as functors, limits as terminal cones, and colimits as initial cocones. Products, coproducts, pullbacks, and pushouts are instances of this common language. The category-of-elements and density constructions show how complex presheaves can be assembled canonically from representables.

The category $\mathbf{Set}$ has all small limits and colimits, as do presheaf categories. This guarantees universal objects in those ambient categories; it does not guarantee that a limit or colimit of abstract model maps is itself a trainable Transformer, nor that it is statistically useful. Section II uses these universal patterns to formulate compositional learning architectures and then states separately what their computational realizations establish.

Suggested Reading

Limits, colimits, and universal properties are developed in [Mac Lane \[1971\]](#) and [Riehl \[2017\]](#). [Fong and Spivak \[2018\]](#) provides diagram-rich applied examples, while Kan's original paper [\[Kan, 1958\]](#) places these constructions in the lineage leading to Kan extensions.

Section II

Compositional Learning

Categorical Deep Learning

Section I supplied the language of categories, functors, representables, and universal constructions. This chapter asks how much of ordinary supervised learning can be expressed compositionally. We first introduce symmetric monoidal categories, then distinguish several possible categorical models of Transformers and language distributions, and finally review the functorial learner construction of Fong, Spivak, and Tuyéras [Fong et al., 2019].

The ambient categories used here are mathematical design choices. Neither a trained Transformer nor the class of Transformer-realizable maps is automatically a topos, a presheaf category, or an enriched category. Each model below therefore states the extra objects, morphisms, and closure laws it requires. Chapter II then adds diagrammatic penalties to a numerical learner; it does not alter the standard definition of backpropagation.

Symmetric Monoidal Category

We now equip a category with a tensor product $c \otimes d$. String concatenation is a monoidal example, but it is generally not symmetric because word order matters. The symmetric structure used for compositional learners instead models independent components that may be exchanged coherently. This distinction prevents the symmetry of a modeling category from erasing the ordered syntax of language.

Symmetric Monoidal Categories

Definition 23. A **monoidal category** is a category C together with a functor $\otimes : C \times C \rightarrow C$, an identity object e of C and natural isomorphisms α, λ, ρ defined as:

$$\begin{aligned}\alpha_{C_1, C_2, C_3} : (C_1 \otimes C_2) \otimes C_3 &\cong C_1 \otimes (C_2 \otimes C_3), \\ \lambda_C : e \otimes C &\cong C, \\ \rho_C : C \otimes e &\cong C,\end{aligned}$$

The natural isomorphisms must satisfy the pentagon and triangle coherence axioms. Mac Lane’s coherence theorem states that diagrams assembled

solely from associators, unitors, identities, tensor products, and their inverses commute; it does not say that arbitrary diagrams in a monoidal category commute [Mac Lane, 1971, Chapter VII]. There are many natural examples of monoidal categories, including finite sets under Cartesian product, sets and relations under Cartesian product, and suitable categories of Hilbert spaces under tensor product. A Markov category is a symmetric monoidal category with a commutative comonoid structure on every object satisfying the Markov compatibility axioms; in particular its tensor unit I is terminal, so every object X has a unique discard morphism $!_X : X \rightarrow I$ [Fritz, 2020].

Definition 24. A **symmetric monoidal category** is a monoidal category $(\mathcal{C}, \otimes, e, \alpha, \lambda, \rho)$ together with a natural isomorphism

$$\tau_{C_1, C_2} : C_1 \otimes C_2 \cong C_2 \otimes C_1, \text{ for all objects } C_1, C_2$$

where τ is involutive, $\tau_{C_2, C_1} \circ \tau_{C_1, C_2} = 1_{C_1 \otimes C_2}$, and is compatible with the associator and unitors through the standard hexagon and unit coherence axioms.

The τ operator is often drawn as a swap. Some applications use enriched categories, but enrichment is additional structure rather than an automatic property of an AI model.

Definition 25. A **$\mathcal{V}$ -enriched category**, for a monoidal category $(\mathcal{V}, \otimes, I)$, assigns a hom-object $\mathcal{C}(x, y) \in \mathcal{V}$ to each pair of objects, together with composition and identity morphisms in $\mathcal{V}$ satisfying enriched associativity and unit laws. When $\mathcal{V}$ is a commutative monoidal preorder, these reduce to

- $I \leq \mathcal{C}(x, x)$
- $\mathcal{C}(y, z) \otimes \mathcal{C}(x, y) \leq \mathcal{C}(x, z)$

Attention in Transformers

Central to a Transformer is the computation of attention scores [Chaudhari et al., 2021]. Queries and keys need not interact symmetrically, because they are produced by different learned maps. Without positional information, self-attention is permutation equivariant in the token axis; positional encodings or position-dependent masks deliberately modify that symmetry. The following equations describe the numerical operation before any categorical interpretation is imposed.

The general structure of an attention model is given as

$$A(q, K, V) = \sum_i \text{softmax}_i(a(q, k_i)) v_i.$$

where K and V collect keys and values, q is a query, a is an alignment function, and softmax normalizes scores over the key index. Some sample alignment functions are given in the following table.

Function	Equation
similarity	$a(k_i, q) = \text{sim}(k_i, q)$
dot product	$a(k_i, q) = q^T k_i$
scaled dot product	$a(k_i, q) = \frac{q^T k_i}{\sqrt{d_k}}$
general	$a(k_i, q) = q^T W k_i$
biased general	$a(k_i, q) = k_i(Wq + b)$
activated general	$a(k_i, q) = \text{act}(q^T W k_i + b)$
generalized kernel	$a(k_i, q) = \phi(q)^T \phi(k_i)$

Table 1: Some Alignment Functions used in Transformer models.

Transformers as a Category

To obtain an actual category, we use representation spaces as objects and equivariant maps as morphisms. Fix a sequence length n . For feature dimensions d, e , a morphism $f : \mathbb{R}^{n \times d} \rightarrow \mathbb{R}^{n \times e}$ is permutation equivariant when

$$f(PX) = Pf(X)$$

for every $n \times n$ permutation matrix P . Identities and composites of equivariant maps are equivariant, so these spaces and maps form a category. Position-free Transformer blocks supply examples of such morphisms; position-dependent encodings and masks require a category that also carries the positional structure.

$$\begin{array}{ccccc}
 X & \xrightarrow{f} & Y & \xrightarrow{g} & Z \\
 \downarrow P & & \downarrow P & & \downarrow P \\
 PX & \xrightarrow{f} & PY & \xrightarrow{g} & PZ
 \end{array}$$

The square states that applying f after permuting token positions gives the same result as permuting the output of f . The larger rectangle records that equivariance is preserved by composition. This is an exact categorical fact about the chosen morphism class, not a claim about every practical Transformer implementation.

Definition 26. A position-free Transformer block maps $X \in \mathbb{R}^{n \times d}$ to an array of the same shape. For head i ,

$$Q_i = XW_Q^i, \quad K_i = XW_K^i, \quad V_i = XW_V^i,$$

$$H_i = \text{softmax}\left(\frac{Q_i K_i^T}{\sqrt{d_h}}\right) V_i.$$

With row-wise softmax, multi-head attention and the feed-forward update may be written schematically as

$$Y = X + \text{Concat}(H_1, \dots, H_h)W_O, \quad Z = Y + \sigma(YW_1 + b_1)W_2 + b_2,$$

omitting normalization and dropout. Shared row-wise linear maps and attention make this block permutation equivariant in the absence of position-dependent terms.

Transformer blocks are therefore a distinguished parameterized family inside the ambient equivariant-map category. The ambient category is closed under composition by definition; a fixed-width or fixed-depth architecture family need not be.

Definition 27. *The category $\mathcal{C}_T$ of Transformer models:*

- *Objects are representation spaces $\mathbb{R}^{n \times d}$ for admissible feature dimensions d .*
- *The arrows or morphisms of the category $\mathcal{C}_T$ are defined as a family of sequence-to-sequence functions and defined as:*

$$\mathcal{C}_T(d, e) = \{f : \mathbb{R}^{n \times d} \rightarrow \mathbb{R}^{n \times e} \mid f(PX) = Pf(X)\}.$$

Composition is ordinary function composition and identities are identity maps. Particular choices of heads and hidden width parameterize subclasses of these morphisms.

LLM Next-token Distributions as Categories

Now, we define categories for LLMs based on next-token distributions, and introduce a hybrid category based on the k -NN LLM model proposed by¹⁴. Raw conditional probabilities do not automatically define an enriched category: identities and enriched composition must be checked. A valid construction begins with the weighted graph of one-token extensions and then takes its enriched path closure.

Definition 28. *Let $\mathcal{V} = ([0, 1], \leq, \cdot, 1)$ and let strings be vertices of a directed graph. Give an edge $x \rightarrow y$ weight $w(x, y) = P(y \mid x)$ when y is an allowed one-token extension of x . The **LLM path category** $\mathcal{L}$ has these strings as objects and hom-value*

$$\mathcal{L}(x, y) = \sup_{\pi: x \rightsquigarrow y} \prod_{e \in \pi} w(e),$$

with the empty path assigned weight 1. Concatenation of paths gives $\mathcal{L}(y, z)\mathcal{L}(x, y) \leq \mathcal{L}(x, z)$, so $\mathcal{L}$ is a $\mathcal{V}$ -enriched category.

Enriched copresheaves $\mathcal{L} \rightarrow \mathcal{V}$ record weighted profiles of possible continuations. The enriched Yoneda image $\mathcal{L}(x, -)$ is one such profile. Calling this a semantic representation is a modeling proposal: the construction records continuation behavior, while semantic adequacy must be evaluated independently.

¹⁴ Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models, 2020. URL <https://arxiv.org/abs/1911.00172>

Definition 29. For the LLM path category $\mathcal{L}$, define the proposed **continuation profile category** as the enriched functor category $[\mathcal{L}, \mathcal{V}]$ of $\mathcal{V}$ -enriched copresheaves and enriched natural transformations.

Finally, let us define a new type of category based on the k -NN large language model.

Definition 30. The k -NN LLM path category $\mathcal{L}_{kNN}$ is obtained by first assigning one-step edge weights

$$p(y|x) = \lambda p_{KNN}(y|x) + (1 - \lambda) p_{LM}(y|x)$$

computed by querying a datastore $\langle \mathcal{K}, \mathcal{V} \rangle$ with an internal self-attention function computed by the Transformer model $f(x)$ to retrieve its k nearest neighbors $\mathcal{N}$ using some distance function $d(., .)$, and using a softmax of their negative distances:

$$p_{KNN}(y|x) \propto \sum_{\langle k_i, v_i \rangle \in \mathcal{N}} \mathbf{1}_{y=v_i} \exp(-d(k_i, f(x)))$$

and then taking the same max-product enriched path closure as above. Its continuation profiles form the enriched functor category $[\mathcal{L}_{kNN}, \mathcal{V}]$.

k -NN language models can improve factual recall in settings studied by Khandelwal et al. [2020]. An analogous category of training processes requires more than declaring that one model “can be modified” into another. One valid choice takes model states as objects and finite, typed training programs as morphisms; the identity is the do-nothing program and composition is sequential execution. RLHF programs can then form a distinguished morphism class if their interfaces are closed under that composition.

Synthetic Token Probabilities using Nearest-Neighbor Language Modeling

k -NN nearest neighbor language models combine a *synthetic probability* constructed from key-query value pairs $\langle k_i, v_i \rangle$ from each training example $\langle c_i, w_i \rangle$, where c_i is a *context-phrase*, such as Charles Dickens wrote, and $f(.)$ is a function that maps a context c to a fixed-length vector representation computed by a pre-trained Transformer model. For example, $f(c)$ could map c to a fixed-length vector output by some self-attention layer (which we described in the previous section). The datastore $\langle \mathcal{K}, \mathcal{V} \rangle$ is the set of all key-value pairs constructed from all the training examples in $\mathcal{D}$:

$$\langle \mathcal{K}, \mathcal{V} \rangle = \{(f(c_i), w_i) | (c_i, w_i) \in \mathcal{D}\}$$

At the time of testing, the language model learned by an LLM generates the output distribution $p_{LM}(y|x)$, and queries the datastore $\langle \mathcal{K}, \mathcal{V} \rangle$ with $f(x)$ to retrieve its k nearest neighbors $\mathcal{N}$ using some distance function $d(., .)$. A

synthetic probability is then computed over the neighbors using a softmax of their negative distances:

$$p_{KNN}(y|x) \propto \sum_{\langle k_i, v_i \rangle \in \mathcal{N}} \mathbf{1}_{y=v_i} \exp(-d(k_i, f(x)))$$

The final distribution for the k -NN LLM is then given by

$$p(y|x) = \lambda p_{KNN}(y|x) + (1 - \lambda) p_{LM}(y|x)$$

Thus, we can define a new LLM category for k -NN LLMs in this way called $\mathcal{L}_{knn}$ where the next-token probabilities are a linear interpolation of the learned probabilities combined with a nearest-neighbor prediction.

Backpropagation as a Functor: Compositional Learning

Our goal is to review the categorical framework of [Fong et al. \[2019\]](#). For a suitable error function and learning rate, their construction maps differentiable parameterized functions to learners and preserves sequential and parallel composition. This is a precise functoriality theorem about the specified categories; it is not the claim that every optimizer, software implementation, or training run is functorial.

Category of Supervised Learning

Functors can be used to give an elegant characterization of the well-known backpropagation algorithm that serves as the “workhorse” of deep learning as a functor over symmetric monoidal categories. In such categories, objects can be “multiplied”: for example, sets form a symmetric monoidal category as the Cartesian product of two sets defines a multiplication operator. A detailed set of coherence axioms are defined for monoidal categories ensure that multiplication is associative, as well as that there are identity operators such that $I \otimes A \simeq A$ for all objects A , where I is the identity object.

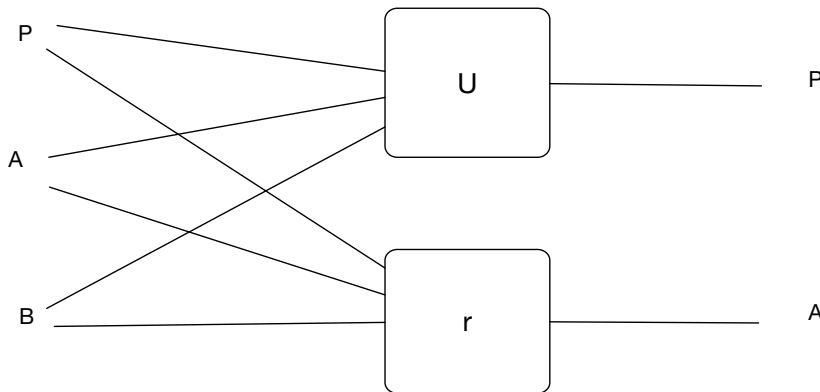


Figure 11: A learner in the symmetric monoidal category Learn is defined as a morphism.

Definition 31. *The symmetric monoidal category **Learn** has sets as objects. A morphism $A \rightarrow B$ is an equivalence class of learners represented by a four-tuple (P, I, U, r) as in [Fong et al. \[2019, Definition II.1 and Proposition II.4\]](#):*

- A parameter space P
- An implementation function $I : P \times A \rightarrow B$
- An update function $U : P \times A \times B \rightarrow P$
- A request function $r : P \times A \times B \rightarrow A$

Note that it is the request function that allows learners to be composed, as each request function transmits information back upstream to earlier learners what output they could have produced that would be more “desirable”. This algebraic characterization of the backpropagation algorithm clarifies its essentially compositional nature

Two learners (P, I, U, r) and (P', I', U', r') are equivalent if there is a bijection $f : P \rightarrow P'$ compatible with all four components, as follows.

- $I'(f(p), a) = I(p, a)$.
- $U'(f(p), a, b) = f(U(p, a, b))$.
- $r'(f(p), a, b) = r(p, a, b)$

Typically, in generative AI trained with neural networks, the parameter space $P = \mathbb{R}^N$ where the neural network has N parameters. The implementation function I represents the “feedforward” component, and the request function represents the “backpropagation” component. The update function represents the change in parameters as a result of processing a training example $(a, f(a)) \in A \times B$.

Note each learner can be combined in sequentially and in parallel, both formally using the operations of composition $\circ$ and tensor product $\otimes$ in the symmetric monoidal category **Learn**, and equivalently in terms of string diagrams. For clarity, let us write out the compositional rule for a pair of learners

$$A \xrightarrow{(P, I, U, r)} B \xrightarrow{(Q, J, V, s)} C$$

The composite learner $A \rightarrow C$ is defined as $(P \times Q, I \cdot J, U \cdot V, r \cdot s)$, where the composite implementation function is

$$(I \cdot J)(p, q, a) := J(q, I(p, a))$$

and the composite update function is

$$(U \cdot V)(p, q, a, c) := (U(p, a, s(q, I(p, a), c)), V(q, I(p, a), c))$$

and the composite request function is

$$(r \cdot s)(p, q, a, c) := r(p, a, s(q, I(p, a), c)).$$

Backpropagation as a Functor

The backpropagation construction is a functor from a category of parameterized differentiable maps to **Learn**. A neural architecture can first be interpreted as a parameterized map and then sent to its learner; this gives the factorization illustrated below. Chapter II adds a separate family of diagram-indexed objectives.

$$\begin{array}{ccc} \mathbf{NNet} & \xrightarrow{\quad} & \mathbf{Learn} \\ & \searrow F & \nearrow L_{\epsilon, e} \\ & \mathbf{Para} & \end{array}$$

Definition 32. The category **Para** has Euclidean spaces as objects. A morphism $A \rightarrow B$ is represented by a Euclidean parameter space P and a differentiable implementation $I : P \times A \rightarrow B$, modulo the equivalence used by [Fong et al. \[2019, Definition III.1\]](#), with one harmless strengthening: (P, I) and (P', I') are equivalent when there is a diffeomorphism $f : P \rightarrow P'$ such that $I'(f(p), a) = I(p, a)$. If $(P, I) : A \rightarrow B$ and $(Q, J) : B \rightarrow C$, their composite is

$$(P \times Q, I \cdot J) \text{ where } (I \cdot J)(p, q, a) = J(q, I(p, a))$$

The monoidal product of objects is Cartesian product (identified with $\mathbb{R}^{n+m}$ for Euclidean spaces). For morphisms $(P, I) : A \rightarrow B$ and $(Q, J) : C \rightarrow D$, the parallel product is $(P \times Q, I \parallel J) : A \times C \rightarrow B \times D$, where

$$(I \parallel J)(p, q, a, c) = (I(p, a), J(q, c))$$

The symmetry is represented by the parameter-free swap $\sigma : A \times B \rightarrow B \times A$, $\sigma(a, b) = (b, a)$.

The cited paper says “differentiable bijection.” Requiring a differentiable inverse as well makes symmetry of the reparameterization relation immediate; the backpropagation construction and its compositional formulas are unchanged.

The backpropagation algorithm can itself be defined as a functor over symmetric monoidal categories

$$L_{\epsilon, e} : \mathbf{Para} \rightarrow \mathbf{Learn}$$

where $\epsilon > 0$ is a real number defining the learning rate for backpropagation, and $e(x, y) : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ is a differentiable error function such that $\frac{\partial e}{\partial x}(x_0, -)$ is invertible for each $x_0 \in \mathbb{R}$. More precisely, [Fong et al. \[2019\]](#),

Theorem III.2] prove that this construction is a faithful, injective-on-objects, strong symmetric monoidal functor. It defines an update procedure for each parameter in a compositional learner. In other words, the functor $L_{\epsilon, e}$ sends each parameterized function $I : P \times A \rightarrow B$ to the learner (P, I, U_I, r_I)

$$U_I(p, a, b) := p - \epsilon \nabla_p E_I(p, a, b)$$

$$r_I(p, a, b) := f_a(\nabla_a E_I(p, a, b))$$

where $E_I(p, a, b) := \sum_j e(I_j(p, a), b_j)$ and f_a is a component-wise application of the inverse to $\frac{\partial e}{\partial x}(a_i, -)$ for each i .

Note that we can easily define functors that define other ways of doing parameterized updates, such as a stochastic approximation method that updates each parameter using only the (noisy) value of the function at the current value of the parameter, and uses a gradual decay of the learning parameters to “simulate” the process of taking gradients. These sort of stochastic approximation updates are now called “zeroth-order” optimization in the deep learning literature.

Summary

This chapter showed how a specified backpropagation construction can be functorial between categories of parameterized maps and learners. That theorem concerns composition of learning systems; it does not force independently chosen paths in an application diagram to agree. Chapter II introduces path-comparison residuals for applications that declare such equations. The additional residual is a task-level structural objective, not what makes ordinary backpropagation itself “truly” functorial.

There are many extensions of the basic framework introduced here. For example, one can try to extend it to other types of machine learning.¹⁵ We will see later that many other types of AGI applications, such as causal inference, can be studied using the principle of functoriality, combined with specific “nice” categories such as toposes.

¹⁵ Design a functor for deep reinforcement learning, and think of why it might lead to a useful way to design novel deep RL methods.

Suggested Reading

Fong et al. [2019] gives the foundational functorial account of backpropagation, and Fong and Spivak [2018] supplies a broader language for compositional systems. Fritz [2020] shows how probabilistic structure is expressed diagrammatically, while Mahadevan [2024a] develops the higher-order architecture pursued later in this book.

Diagrammatic Backpropagation

Diagrammatic Backpropagation (DB) augments an ordinary task loss with residuals derived from declared path equations. If a diagram contains $A \xrightarrow{f} B \xrightarrow{g} C$ and $A \xrightarrow{h} C$ with the requirement $h = g \circ f$, DB penalizes disagreement between the learned direct and composite paths. Gradients of this scalarized objective are computed by ordinary automatic differentiation.

The term *diagrammatic curvature* is used here for these numerical path residuals. It should not be confused with a general categorical curvature invariant. Nor does a vanishing residual alone establish a categorical limit or colimit: commutativity is a compatibility condition, whereas universality also requires quantified existence and uniqueness of factorization maps. DB is therefore a method for enforcing selected diagram equations. Connections to simplicial horn filling and universal constructions are design interpretations that require additional witnesses when asserted literally.

Introduction

The nerve of a category is a simplicial set, and more general simplicial sets provide a combinatorial language for vertices, edges, triangles, and higher-dimensional compatibility data [May, 1992]. DB uses a finite diagram or simplicial skeleton as an index for constraints; the backpropagation functor reviewed in Chapter II remains the optimizer construction. The later Geometric Transformer experiments test whether this extra structural supervision helps on selected compositional benchmarks [Mahadevan, 2024a]. They do not by themselves establish a general explanation of LLM reasoning failures.

Minimizing Diagrammatic Curvature Energy

DIAGRAMMATIC CURVATURE ENERGY is the name given to the sum of chosen path-comparison penalties. In conventional supervised learning, training data are pairs (x, y) and a model minimizes $\mathcal{L}(f_\theta(x), y)$. Such a loss need not compare two routes through a declared diagram unless those comparisons are included in the objective.

DB assumes that a task designer or extraction procedure supplies objects,

morphisms, and equations among selected paths. The nerve construction sends a category $\mathcal{C}$ to a simplicial set $N(\mathcal{C})$, with objects as 0-simplices, composable arrows as 1-dimensional data, and composable strings with their composites represented in higher degrees. In the Triangles benchmark, linear maps $f : A \rightarrow B$, $g : B \rightarrow C$, and $h : A \rightarrow C$ are declared to satisfy $h = g \circ f$. DB augments the task loss with the squared discrepancy between these routes. This enforces the sampled equation; a separate factorization test would be needed to claim approximation of a limit or colimit from Chapter I.

ALGORITHM 1: SIMPLICIAL MESSAGE PASSING IN DB. DB propagates information through a simplicial complex representing a diagram of objects and morphisms. Each node corresponds to an object, and each directed edge corresponds to a morphism equipped with a learned linear map M_{ij} and relation embedding r_{ij} . At every iteration, a node aggregates messages arriving from its incident edges, each message combining the transformed neighbor representation $M_{ij}x_j$, a relation embedding, and any residual features that capture local curvature. The messages are degree-normalized and used to update the node state through a gated recurrent cell. Normalizing transformed messages controls one source of scale variation, although stability of the complete recurrent system remains an empirical property. The process diffuses residual information through the finite constraint graph, so node embeddings can evolve toward agreement on the paths that were actually registered. It does not test unregistered composites or the existence and uniqueness clauses of a universal property.

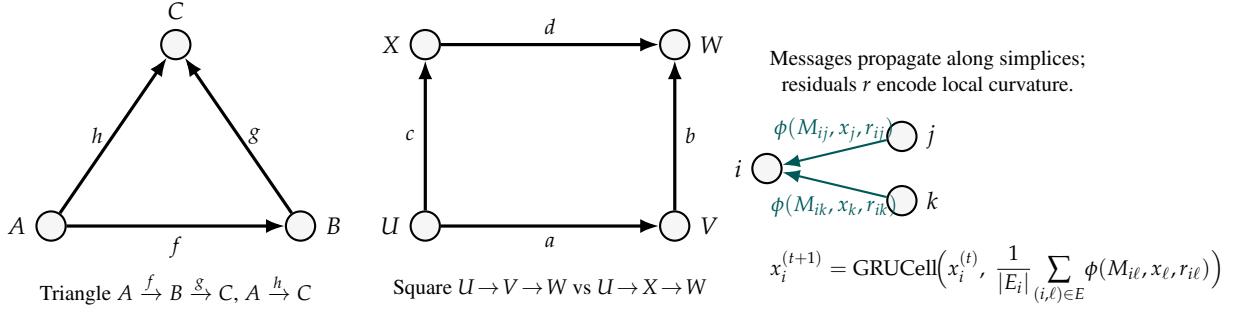
Algorithm 1: Simplicial Message Passing in DB

Input: Diagram $\mathcal{D} = (V, E)$ with objects V and morphisms E , node states $x_i^{(0)} \in \mathbb{R}^d$, edge maps $M_{ij} \in \mathbb{R}^{d \times d}$, relation ids r_{ij} , residual features ρ_{ij} , steps T

Output: Refined node states $x_i^{(T)}$

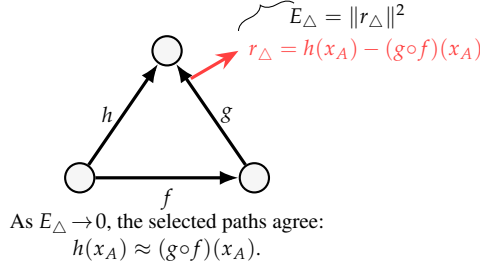
```

1: for  $t = 0, 1, \dots, T - 1$  do
2:   for each node  $i \in V$  do
3:      $m_i \leftarrow 0$ 
4:     for each incoming edge  $(i \leftarrow j) \in E$  do
5:        $z \leftarrow M_{ij}x_j^{(t)}$ ;  $z \leftarrow z / (\|z\|_2 + \epsilon)$  ▷ normalize edge
       transform
6:        $u \leftarrow \phi(z, r_{ij}, \rho_{ij})$  ▷ e.g.  $u = \tanh(W_e[z; \rho] + b_e) + \text{Emb}(r)$ 
7:        $m_i \leftarrow m_i + u$ 
8:     end for
9:      $m_i \leftarrow \frac{1}{\max(1, |\{j: (i \leftarrow j) \in E\}|)} m_i$  ▷ degree-normalize messages
10:     $x_i^{(t+1)} \leftarrow \text{GRUCell}(x_i^{(t)}, m_i)$  ▷ curvature diffuses via messages
11:  end for
12: end for
```

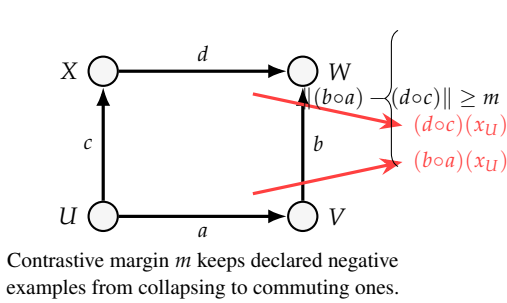


(a) **Simplicial diagrams.** Edges carry morphisms $M_f, M_g, \dots$

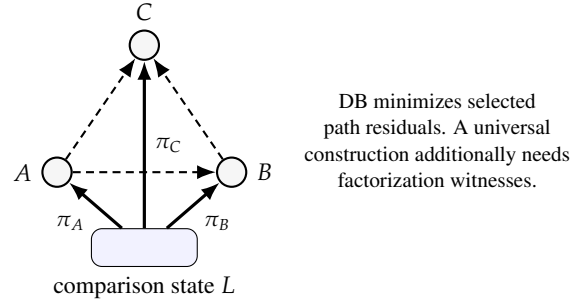
(b) **Simplicial message passing.** Curvature diffuses via messages.



(c) **Path compatibility.** Triangle energy measures disagreement on the selected probe.



(d) **Negative-example separation.** A margin preserves a task-defined distinction; it is not a colimit certificate.



(e) **Structural view.** Compatibility and universality are distinct obligations.

Figure 12: Simplicial message passing and path compatibility in DB. (A) Triangle and square simplices encode compositional constraints. (B) DB aggregates messages along simplices; residuals carry local curvature. (C) Triangle energy measures a declared path equation on sampled probes. (D) A margin separates declared negative diagrams. Neither condition alone certifies a categorical limit or colimit.

ALGORITHM 2: LEARNED COMPOSITIONAL ENERGY. This routine computes the *diagrammatic curvature energy* $E_\theta(\mathcal{D})$ from the learned path embeddings. In the algorithm, ET_θ represents $\text{edge\_transform}_\theta$. For each triangle constraint, the algorithm measures the discrepancy between the direct morphism h and the composed morphism $g \circ f$. For each square, it compares the two-path compositions $(b \circ a)$ and $(d \circ c)$. These residuals are squared and accumulated to yield E_Δ and $E_\square$. The resulting energy measures disagreement for the registered triangles and squares on the evaluated node states. $E_\theta(\mathcal{D}) = 0$ means those finite residuals vanish; it does not imply equality of the underlying maps on all inputs unless the probe set is separating. Minimizing the energy is thus a compatibility-seeking objective. A “limit-seeking” interpretation is valid only when separate factorization and uniqueness witnesses are supplied.

Algorithm 2: Learned Compositional Energy $E_\theta(\mathcal{D})$

Input: Triangle constraints $\mathcal{T}$, square constraints $\mathcal{S}$, node states x , maps M , relations r

Output: $E_\theta(\mathcal{D}) = E_\Delta + E_\square$

```

1:  $E_\Delta \leftarrow 0, E_\square \leftarrow 0$ 
2: for each triangle  $(A \xrightarrow{f} B \xrightarrow{g} C, A \xrightarrow{h} C) \in \mathcal{T}$  do
3:    $z_h \leftarrow \text{ET}_\theta(M_h, r_h, x_A)$ 
4:    $z_{gf} \leftarrow \text{ET}_\theta(M_g, r_g, \text{ET}_\theta(M_f, r_f, x_A))$ 
5:    $E_\Delta \leftarrow E_\Delta + \|z_h - z_{gf}\|_2^2$ 
6: end for
7: for each square  $(U \xrightarrow{a} V \xrightarrow{b} W, U \xrightarrow{c} X \xrightarrow{d} W) \in \mathcal{S}$  do
8:    $z_{ba} \leftarrow \text{ET}_\theta(M_b, r_b, \text{ET}_\theta(M_a, r_a, x_U))$ 
9:    $z_{dc} \leftarrow \text{ET}_\theta(M_d, r_d, \text{ET}_\theta(M_c, r_c, x_U))$ 
10:   $E_\square \leftarrow E_\square + \|z_{ba} - z_{dc}\|_2^2$ 
11: end for
12: return  $E_\Delta + E_\square$ 

```

ALGORITHM 3: RAW ENERGY VECTOR. The raw energy computation isolates a structural signature derived solely from the exogenous morphism matrices M_f . It measures the same triangle and square residuals as Algorithm 2 but without involving the learnable message-passing parameters. The result is a two-dimensional feature vector $[E_\Delta^{\text{raw}}, E_\square^{\text{raw}}]$ that can feed a lightweight classifier. It depends directly on the supplied maps and constraints and is therefore a task-specific diagnostic, not a learned discovery of structure. If labels are defined by corruption of those same residuals, a raw-energy classifier can solve the benchmark without message passing; it must consequently be reported as a baseline or ablation rather than evidence for the learned DB component.

Algorithm 3: Raw Energy Vector $E^{raw}(\mathcal{D}) = [E_{\triangle}^{raw}, E_{\square}^{raw}]$

Input: Constraints $(\mathcal{T}, \mathcal{S})$, node states x , exogenous maps M

Output: $[E_{\triangle}^{raw}, E_{\square}^{raw}]$

```

1:  $E_{\triangle}^{raw} \leftarrow 0, E_{\square}^{raw} \leftarrow 0$ 
2: for each triangle  $(f, g, h, A)$  in  $\mathcal{T}$  do
3:    $E_{\triangle}^{raw} \leftarrow E_{\triangle}^{raw} + \|M_h x_A - (M_g M_f) x_A\|_2^2$ 
4: end for
5: for each square  $(a, b, c, d, U)$  in  $\mathcal{S}$  do
6:    $E_{\square}^{raw} \leftarrow E_{\square}^{raw} + \|M_b M_a - M_d M_c\|_F^2$ 
7: end for
8: return  $[E_{\triangle}^{raw} / \max(1, |\mathcal{T}|), E_{\square}^{raw} / \max(1, |\mathcal{S}|)]$ 
    
```

ALGORITHM 4: DB TRAINING LOOP. This is the unifying optimization routine that integrates message passing, energy minimization, and contrastive margin control. For each batch of diagrams, DB first computes the raw and learned energies, then classifies each sample using the raw-energy head. The total loss combines a cross-entropy classification term with the geometric energy terms. Positive diagrams receive a direct curvature penalty proportional to E_{θ} , encouraging exact commutativity; negatives are trained with a contrastive hinge that enforces a minimum energy margin m , preventing degenerate flattening. A warm-up schedule for λ_{geom} allows the classifier to stabilize before the geometric term dominates, mirroring a gradual transition from syntactic to semantic consistency. The optimizer (AdamW with weight decay and gradient clipping) updates both the message-passing parameters and the head, while normalization of the edge transforms ensures bounded activations. Collectively, this loop drives positive examples toward low registered path residual and keeps negative examples beyond a chosen margin. The latter is a contrastive classification device, not a numerical realization of a colimit.

Algorithm 4: DB Training Loop

Input: Dataset $\{\mathcal{D}_i, y_i\}$ with diagrams and labels, steps T , warmup T_w , hyperparams $(\lambda_{\text{geom}}, \lambda_{\text{contr}}, m)$

Output: Parameters θ (message-passing + edge transforms) and head ψ

```

1: for  $t = 1, \dots, T$  do
2:   for minibatch  $\mathcal{B}$  do
3:      $\mathcal{L} \leftarrow 0$ 
4:     for each  $(\mathcal{D}, y) \in \mathcal{B}$  do
5:        $E^{\text{raw}} \leftarrow \text{Alg. 3}; E_\theta \leftarrow \text{Alg. 2}$ 
6:        $\hat{y} \leftarrow \text{Head}_\psi(E^{\text{raw}})$   $\triangleright$  small MLP on  $[E_{\triangle}^{\text{raw}}, E_{\square}^{\text{raw}}]$ 
7:        $\mathcal{L} \leftarrow \mathcal{L} + \text{CE}(\hat{y}, y)$ 
8:        $\lambda_g \leftarrow \lambda_{\text{geom}} \cdot \min(1, t/T_w)$   $\triangleright$  geometric warm-up
9:       if  $y = 1$  then  $\triangleright$  positives: enforce registered equations
10:         $\mathcal{L} \leftarrow \mathcal{L} + \lambda_g E_\theta$ 
11:       else  $\triangleright$  negatives: maintain a contrastive margin
12:         $\mathcal{L} \leftarrow \mathcal{L} + \lambda_{\text{contr}} \cdot \max(0, m - E_\theta)^2$ 
13:       end if
14:     end for
15:     Update  $(\theta, \psi)$  by AdamW + clipping; normalize edge trans-
        forms
16:   end for
17: end for
```

ALGORITHM 5: CONSTRAINT EXTRACTION FROM DIAGRAM. This preprocessing step converts each diagram or algebraic problem instance into a structured set of triangle and square constraints. It scans all triples of connected morphisms to identify potential triangles $A \xrightarrow{f} B \xrightarrow{g} C$ with a direct shortcut $A \xrightarrow{h} C$, and all quadruples forming squares $U \xrightarrow{a} V \xrightarrow{b} W$ versus $U \xrightarrow{c} X \xrightarrow{d} W$. These lists define the finite skeleton over which DB performs message passing and energy evaluation. The representation is reusable across domains only when a sound compiler supplies typed objects, maps, and equations. Constraint extraction by graph pattern alone discovers candidate parallel paths; it does not establish that the paths ought to be equal.

 Algorithm 5: Constraint Extraction from Diagram

Input: Diagram $\mathcal{D} = (V, E)$ with typed morphisms; anchor rule (e.g. choose A in triangles)
Output: Triangle set $\mathcal{T}$, square set $\mathcal{S}$

- 1: $\mathcal{T} \leftarrow \emptyset, \mathcal{S} \leftarrow \emptyset$
- 2: **for** each triple $(A \xrightarrow{f} B \xrightarrow{g} C)$ with $A \xrightarrow{h} C$ **do**
- 3: $\mathcal{T} \leftarrow \mathcal{T} \cup \{(f, g, h, A)\}$
- 4: **end for**
- 5: **for** each quadruple $(U \xrightarrow{a} V \xrightarrow{b} W, U \xrightarrow{c} X \xrightarrow{d} W)$ **do**
- 6: $\mathcal{S} \leftarrow \mathcal{S} \cup \{(a, b, c, d, U)\}$
- 7: **end for**
- 8: **return** $(\mathcal{T}, \mathcal{S})$

Summary

Diagrammatic Backpropagation is a task-level method for adding declared path equations to an ordinary differentiable objective. Its strongest immediate claim is therefore conditional: when the equations are semantically valid, the residual makes their violation observable and optimizable. The method does not infer those equations automatically, and low residual does not by itself prove a universal construction.

The next chapter instantiates this idea in the Geometric Transformer and evaluates it on selected benchmarks. Those experiments must be read in terms of their stated baselines, datasets, and ablations; they do not establish that DB dominates standard Transformers across language modeling generally.

There are a number of related studies that generalize deep learning using categorical methods, and you can find references to this literature on the web.
 16

Suggested Reading

Functorial backpropagation provides the closest categorical starting point [Fong et al., 2019]. The simplicial language used here is developed classically by May [1992]; for a modern topological-neural formulation, see Hajij et al. [2025]. The larger generative-AI architecture appears in Mahadevan [2024a]. The subsequent LINC framework replaces scalarized error with the diagnosis and infinitesimal repair of non-compositional sketches [Mahadevan, 2026f].

¹⁶ Mustafa Hajij, Lennart Bastian, Sarah Osentoski, Hardik Kabaria, John L. Davenport, Sheik Dawood, Balaji Cherukuri, Joseph G. Kocheemoolayil, Nastaran Shahmansouri, Adrian Lew, Theodore Papamarkou, and Tolga Birdal. Copresheaf topological neural networks: A generalized deep learning framework, 2025. URL <https://arxiv.org/abs/2505.21251>

Section III

Geometric and Kan Extension Transformers

Geometric Transformers

This chapter introduces Geometric Transformers (GTs), a family of neural blocks that augments global attention with explicitly registered local relations. GT-Lite adds a local sequence operator; GT-Full also maintains features on selected edges and triangles and may be trained with the Diagrammatic Backpropagation (DB) residuals introduced in Chapter 6. The categorical contribution is therefore a design discipline: declare a finite relational shape, assign trainable maps to it, and measure selected failures of path compatibility.

Three levels of claim must remain separate. Simplicial sets and the Yoneda lemma supply established mathematical language. Their use to interpret GT is a modeling proposal. Performance is an empirical question that must be answered under matched information budgets. Early language-modeling runs produced spectacularly low perplexities, but later ablations showed that the geometric branch could see non-causal information. Those runs demonstrate the importance of information pathways, not architectural superiority. The strict-causal comparisons reported later in the book show smaller, task-dependent gains. This chapter develops the architecture and explains how that empirical correction changes its interpretation.

A Yoneda-style view of the Geometric Transformer

A SAFE CATEGORICAL READING BEGINS WITH A DECLARATION. Let G be a directed graph whose vertices are token positions and whose generating edges record allowed local relations—for example, causal adjacency or a chosen positional shift. The free category $\mathcal{C} = \text{Free}(G)$ has paths in G as morphisms. Passing to the free category is essential: a raw “can attend to” relation need not itself contain identities or be closed under composition.

For each object $x \in \mathcal{C}$ the *representable presheaf*

$$h_x := \mathcal{C}(-, x) : \mathcal{C}^{\text{op}} \rightarrow \mathbf{Set}$$

assigns to each $z \in \mathcal{C}$ the set of morphisms $z \rightarrow x$. The Yoneda lemma says that

$$\mathcal{C}(x, y) \cong \text{Nat}(\mathcal{C}(-, x), \mathcal{C}(-, y)) = \text{Nat}(h_x, h_y), \quad (1)$$

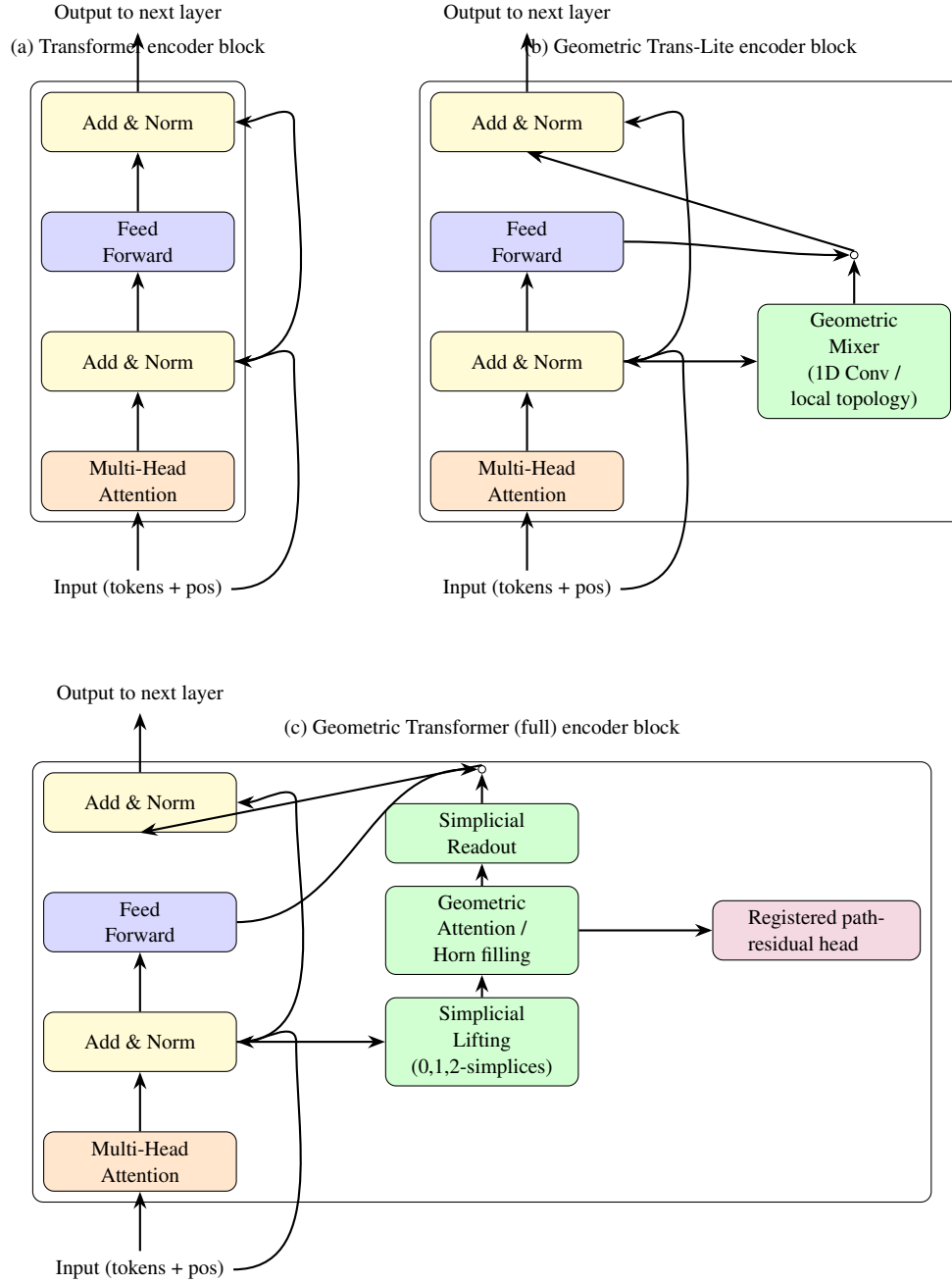


Figure 13: Comparison of encoder blocks. (a) Standard Transformer encoder block. (b) GeomTrans-Lite augments each layer with a local geometric mixer. (c) GT-Full uses a simplicial lifting–mixing–readout stack and an auxiliary head for registered path residuals. The figure specifies a model family, not a theorem that the learned maps form a simplicial object.

i.e., a morphism $x \rightarrow y$ is equivalent to a natural transformation between the representables h_x and h_y .

A Yoneda-style interpretation of attention may then be used as a design analogy:

- a token embedding is treated as a finite numerical profile of the relations incident to x ;
- an attention head compares such profiles and transports value vectors; and
- naturality would be an additional equation saying that these comparisons respect the declared reindexing maps.

An ordinary attention head is therefore not automatically a natural transformation between representables. The Yoneda lemma is exact; a learned attention tensor is finite, parameter-dependent, and generally unconstrained by naturality equations. The useful lesson is narrower: relational profiles can be enriched by recording how local relations compose.

Diagram functors for GT Lite and GT Full

The Geometric Transformer introduces two extra ingredients:

1. a *local geometric path*, implemented as a 1D convolution/smoothing over neighboring tokens (GT-Lite), and
2. a *simplicial / diagrammatic message passing* step, in which information flows not only along single edges, but also around small relational motifs (triangles, horns), together with a curvature term that penalizes inconsistency across such motifs (GT-Full).

Categorically, the intended extension replaces isolated relational profiles by features indexed by a declared family of vertices, edges, and triangles.

A PRECISE INDEXING CATEGORY. The nerve NC is the simplicial set whose n -simplices are strings of n composable morphisms. Its category of simplices $\Delta/(NC)$ has simplices as objects and simplicial operators over NC as morphisms. Let $\text{Simp}_x(C)$ denote the full subcategory on simplices whose final vertex is x . A diagram-valued feature assignment would be a functor

$$D_x : \text{Simp}_x(C)^{\text{op}} \longrightarrow \mathbf{Vect}.$$

This definition states what exact functoriality would mean. A concrete GT implementation usually stores only a finite truncation—commonly vertices, selected edges, and selected triangles—and need not realize every map in D_x .

Intuitively, D_x does not just remember the arrows $z \rightarrow x$ (as h_x does); it remembers entire *local diagrams ending at x* : paths, triangles, and higher simplices. In the implementation:

- the depthwise 1D convolution in GT-Lite performs a learned local aggregation over a fixed neighborhood. It may be compared with a weighted colimit only after an indexing category, a diagram, and a cocone have been specified; the convolution alone is not a coend;
- the simplicial message passing in GT-Full takes as input embeddings for edges and triangles and returns a refined embedding that depends on the entire 1- and 2-simplicial structure around x . This is a finite neural realization of features indexed by a truncated simplicial neighborhood. Calling it a parametrization of D_x is justified only when its restriction maps and functor laws are also supplied.

CURVATURE LOSS AS FUNCTORIALITY CONSTRAINT. The DB loss can penalize a registered path equation across a selected 2-simplex. Concretely, for a triangle

$$z \xrightarrow{g} x, \quad z \xrightarrow{h} y, \quad x \xrightarrow{k} y,$$

one may penalize the difference between the learned realization of h and that of $k \circ g$ on sampled states. This is a local path-compatibility residual. We sometimes call it *discrete curvature*, but that name does not make it an intrinsic curvature invariant, nor does a small residual on a finite probe set prove full functoriality.

Thus GT exposes selected local diagrams to learning, while DB supplies a way to test declared equalities on those diagrams. The distinction between this finite realization and an exact diagram functor will remain explicit.

A diagrammatic Yoneda principle

With this notation in hand, we can state the more modest “GT Yoneda heuristic.”

Yoneda says that an object is determined by its exact representable profile. GT borrows this relational viewpoint, but indexes learned features by selected local simplices rather than only by vertices. If a future construction realizes functors D_x, D_y and a natural transformation $D_x \Rightarrow D_y$, Yoneda-style reasoning can then be applied exactly in the relevant functor category. The architecture used here is a finite approximation to that design target, not an instance of the Yoneda lemma by definition.

At a high level, the difference is:

Vanilla Transformer	Geometric Transformer
learned features primarily on positions	features on positions and selected local simplices
global attention mixing	global mixing plus registered local operators
no explicit path residual	optional residuals on declared parallel paths

This heuristic explains what structural information GT attempts to expose. It does not predict an empirical gain without assumptions about the task, optimization, and information available to each branch. Those variables are separated experimentally below and in Chapter 9.

Diagrammatic Backpropagation as Horn Filling in Simplicial Sets

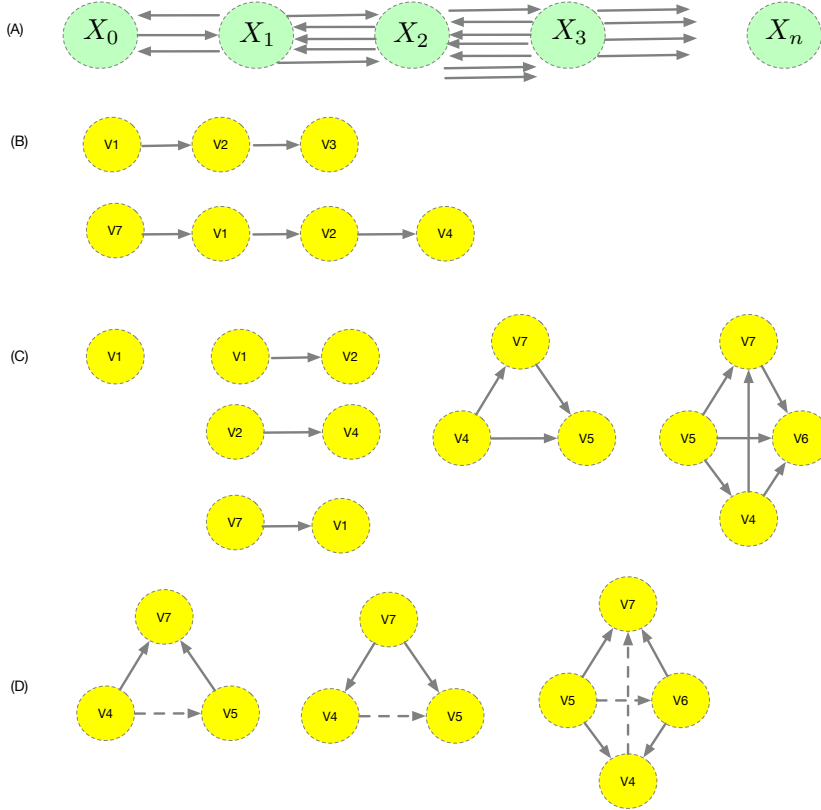
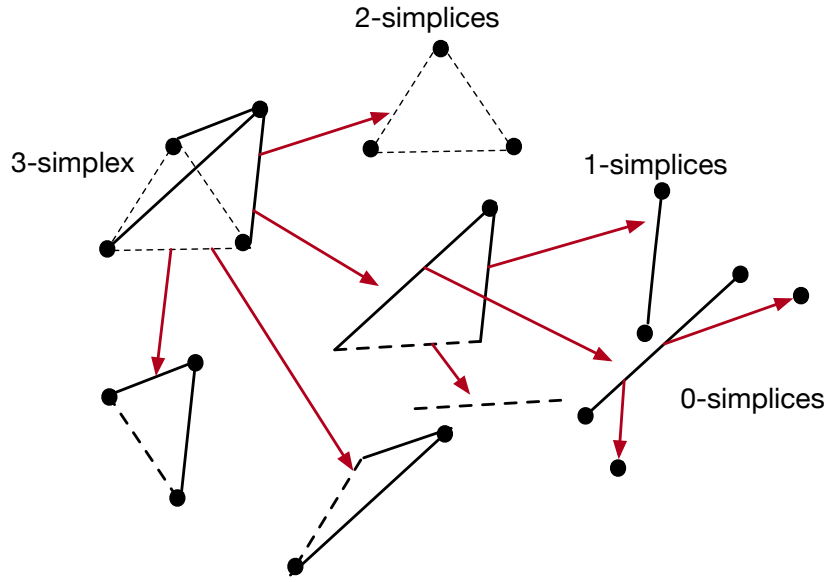


Figure 14: DB and GT as hierarchical simplicial learning. (A) X_0 stores entities, X_1 morphisms, and X_n higher interactions. (B) X_0 and X_1 recover sequential composition. (C) Higher simplices coordinate their faces through hierarchical information flow. (D) Outer-horn extensions pose a harder completion problem than inner-horn extensions.

This section recalls the exact mathematics of simplicial sets and then marks the point at which GT introduces a computational analogy. The nerve functor $N : \mathbf{Cat} \rightarrow \mathbf{sSet}$ is fully faithful: a category can be recovered from its nerve. Its left adjoint, categorification, sends a general simplicial set to a category and is not fully faithful on all simplicial sets. This fact says that general simplicial sets contain data not determined by an ordinary category; it does *not* prove that a simplicial neural architecture is more expressive or learns better.

The simplex category Δ provides the indexing language. A simplicial set is a functor $X : \Delta^{\text{op}} \rightarrow \mathbf{Set}$; more generally, a simplicial object in a category $\mathcal{E}$ is a functor $\Delta^{\text{op}} \rightarrow \mathcal{E}$. Choosing $\mathcal{E}$ to be a category of vector spaces or typed parameterized maps is an additional modeling step. GT uses a finite



Decomposition of a 3-simplex into its parts

Figure 15: A hierarchical design metaphor for GT. Features attached to a 3-simplex can exchange information with features on its faces. Such an update rule must be specified separately; it is not supplied by the definition of a simplicial set.

truncation of this indexing pattern to organize neural features and updates.

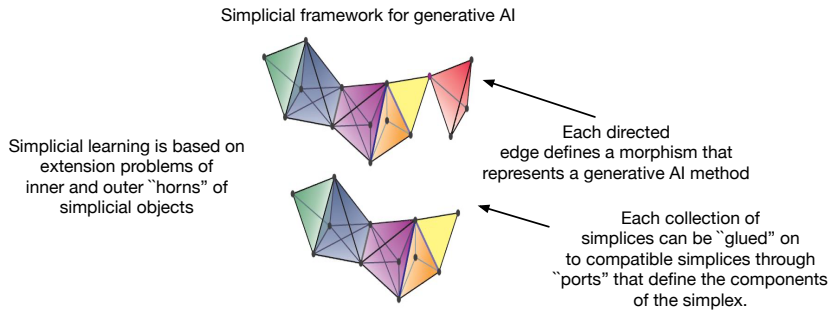


Figure 16: A simplicial indexing scheme for learning components. Vertices, edges, and triangles can carry typed features, while face maps relate each simplex to its boundary data. Backpropagation optimizes a chosen numerical realization; the simplicial identities specify the structural equations that such a realization may be asked to respect.

Simplicial Sets and Objects

A simplicial set consists of sets X_n for $n \geq 0$, face maps $d_i : X_n \rightarrow X_{n-1}$, and degeneracy maps $s_i : X_n \rightarrow X_{n+1}$ satisfying the simplicial identities. The graded sets alone are insufficient. In the nerve of a category, X_0 contains objects, X_1 contains morphisms, and X_n contains strings of n composable morphisms. For a general simplicial set, however, an n -simplex need not be determined by such a string. GT treats selected simplices as indices for features; any trainable update between dimensions is part of the architecture,

not part of the definition above.

The objects of Δ are the nonempty finite ordinals $[n] = \{0, 1, \dots, n\}$ and its morphisms are order-preserving maps. Every such map factors uniquely as a surjection followed by an injection. The elementary injections $\delta^i : [n-1] \rightarrow [n]$ omit i , and the elementary surjections $\sigma^i : [n+1] \rightarrow [n]$ repeat i . The standard simplex is the representable simplicial set $\Delta[n] = \Delta(-, [n])$. Yoneda gives $X_n \cong \mathbf{sSet}(\Delta[n], X)$: an n -simplex of X is equivalently a map from the standard n -simplex.

Any morphism in the category Δ can be defined as a sequence of *co-degeneracy* and *co-face* operators, where the co-face operator $\delta_i : [n-1] \rightarrow [n]$, $0 \leq i \leq n$ is defined as:

$$\delta_i(j) = \begin{cases} j, & \text{for } 0 \leq j \leq i-1 \\ j+1 & \text{for } i \leq j \leq n-1 \end{cases}$$

Analogously, the co-degeneracy operator $\sigma_j : [n+1] \rightarrow [n]$ is defined as

$$\sigma_j(k) = \begin{cases} k, & \text{for } 0 \leq k \leq j \\ k-1 & \text{for } j < k \leq n+1 \end{cases}$$

Note that under the contravariant mappings, co-face mappings turn into face mappings, and co-degeneracy mappings turn into degeneracy mappings. That is, for any simplicial object (or set) X_n , we have $X(\delta_i) := d_i : X_n \rightarrow X_{n-1}$, and likewise, $X(\sigma_j) := s_j : X_{n-1} \rightarrow X_n$.

The induced face and degeneracy maps satisfy the simplicial identities

$$\begin{aligned} d_i d_j &= d_{j-1} d_i & (i < j), \\ s_i s_j &= s_{j+1} s_i & (i \leq j), \\ d_i s_j &= \begin{cases} s_{j-1} d_i, & i < j, \\ \text{id}, & i = j \text{ or } i = j+1, \\ s_j d_{i-1}, & i > j+1. \end{cases} \end{aligned}$$

Example 8. In the nerve of a category $\mathcal{C}$, vertices are objects and edges are morphisms $f : X \rightarrow Y$. The face maps $d_0 f = Y$ and $d_1 f = X$ recover the target and source. The degeneracy map $s_0 X$ is the identity morphism $1_X : X \rightarrow X$.

Example 9. Given a category $\mathcal{C}$, we can identify an n -simplex σ of a simplicial set $\mathcal{C}_n$ with the sequence:

$$\sigma = C_0 \xrightarrow{f_1} C_1 \xrightarrow{f_2} \dots \xrightarrow{f_n} C_n$$

the face operator d_0 applied to σ yields the sequence

$$d_0 \sigma = C_1 \xrightarrow{f_2} C_2 \xrightarrow{f_3} \dots \xrightarrow{f_n} C_n$$

where the object C_0 is deleted along with the morphism f_1 leaving it.

Example 10. Given a category $\mathcal{C}$, and an n -simplex σ of the simplicial set $\mathcal{C}_n$, the face operator d_n applied to σ yields the sequence

$$d_n\sigma = C_0 \xrightarrow{f_1} C_1 \xrightarrow{f_2} \dots \xrightarrow{f_{n-1}} C_{n-1}$$

where the object C_n is deleted along with the morphism f_n entering it. Deleting a face is a simplicial operation; it is not, without additional causal semantics, an intervention on a causal model.

Example 11. Given a category $\mathcal{C}$, and an n -simplex σ of the simplicial set $\mathcal{C}_n$ the face operator $d_i, 0 < i < n$ applied to σ yields the sequence

$$d_i\sigma = C_0 \xrightarrow{f_1} C_1 \xrightarrow{f_2} \dots C_{i-1} \xrightarrow{f_{i+1} \circ f_i} C_{i+1} \dots \xrightarrow{f_n} C_n$$

where the object C_i is deleted and the morphism f_i is composed with f_{i+1} . This is categorical composition in the nerve, not an assignment of a value to C_i .

Example 12. Given a category $\mathcal{C}$, and an n -simplex σ of the simplicial set $\mathcal{C}_n$, the degeneracy operator $s_i, 0 \leq i \leq n$ applied to σ yields the sequence

$$s_i\sigma = C_0 \xrightarrow{f_1} C_1 \xrightarrow{f_2} \dots C_i \xrightarrow{1_{C_i}} C_i \xrightarrow{f_{i+1}} C_{i+1} \dots \xrightarrow{f_n} C_n$$

where the object C_i is “repeated” by inserting its identity morphism 1_{C_i} .

Definition 33. Given a category $\mathcal{C}$, and an n -simplex σ of the simplicial set $\mathcal{C}_n$, σ is a **degenerate** simplex if some f_i in σ is an identity morphism, in which case C_i and C_{i+1} are equal.

Hierarchical Learning in DB and GT by solving Lifting Problems

GT uses a hierarchical model of features indexed by selected simplices. This extends edgewise composition with higher-arity local contexts, but it does not replace ordinary composition. Lifting problems provide a useful exact language for asking whether compatible boundary data admit an extension.

For example, a simplicial vector space is one functor $X : \Delta^{\text{op}} \rightarrow \mathbf{Vect}$, not a separate functor for each simplex. Its face and degeneracy maps satisfy the identities above. A finite GT layer borrows this organization: it attaches tensors to selected simplex types and aggregates along incidence maps. Exact consistency with a simplicial object would have to be verified; DB can register some, but not all, of the required equations.

Lifting problems provide elegant ways to define solutions to computational problems in category theory regarding the existence of mappings. For example, the notion of injective and surjective functions, the notion of separation in topology, and many other basic constructs can be formulated as solutions to lifting problems. Lifting problems define ways of decomposing structures into simpler pieces, and putting them back together again.

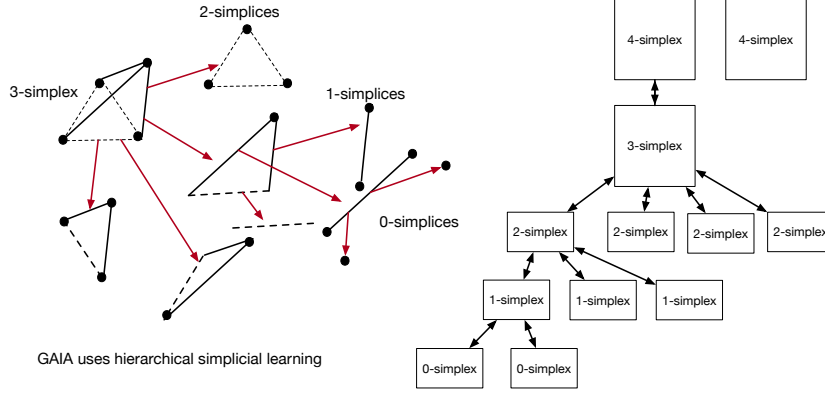


Figure 17: A 3-simplex and its boundary motivate a hierarchy of feature updates. The simplex supplies incidence data; the neural architecture supplies the parameters and update maps.

Definition 34. Let $\mathcal{C}$ be a category. A **lifting problem** is a commutative square of the following form, together with the question of whether a diagonal filler exists.

$$\begin{array}{ccc} A & \xrightarrow{\mu} & X \\ \downarrow f & & \downarrow p \\ B & \xrightarrow{\nu} & Y \end{array}$$

Definition 35. Let $\mathcal{C}$ be a category. A **solution to a lifting problem** in $\mathcal{C}$ is a morphism $h : B \rightarrow X$ in $\mathcal{C}$ satisfying $p \circ h = \nu$ and $h \circ f = \mu$ as indicated in the diagram below.

$$\begin{array}{ccc} A & \xrightarrow{\mu} & X \\ \downarrow f & \nearrow h & \downarrow p \\ B & \xrightarrow{\nu} & Y \end{array}$$

Definition 36. Let $\mathcal{C}$ be a category. Given morphisms $f : A \rightarrow B$ and $p : X \rightarrow Y$, we say that f has the **left lifting property** with respect to p , or that p has the **right lifting property** with respect to f , if every commutative square $p \circ \mu = \nu \circ f$ of the following form

$$\begin{array}{ccc} A & \xrightarrow{\mu} & X \\ \downarrow f & \nearrow h & \downarrow p \\ B & \xrightarrow{\nu} & Y \end{array}$$

admits a solution given by the map $h : B \rightarrow X$ satisfying $p \circ h = \nu$ and $h \circ f = \mu$.

A lifting problem is an existence question for a diagonal morphism. A parameterized optimizer can be designed to search for an approximate diagonal in a numerical realization, but ordinary backpropagation is not thereby a solution to every lifting problem. Two exact examples in **Set** build the relevant intuition.

Example 13. Given the map $f : \emptyset \rightarrow \{\bullet\}$, a function p has the right lifting property with respect to f precisely when p is surjective.

$$\begin{array}{ccc} \emptyset & \xrightarrow{\mu} & X \\ \downarrow f & \nearrow h & \downarrow p \\ \{\bullet\} & \xrightarrow{\nu} & Y \end{array}$$

Similarly, the following lifting problem characterizes injective functions.

Example 14. Given the fold map $f : \{0, 1\} \rightarrow \{\bullet\}$, a function p has the right lifting property with respect to f precisely when p is injective.

$$\begin{array}{ccc} \{\bullet, \bullet\} & \xrightarrow{\mu} & X \\ \downarrow f & \nearrow h & \downarrow p \\ \{\bullet\} & \xrightarrow{\nu} & Y \end{array}$$

Simplicial Subsets and Horns in DB and GT

Horn inclusions provide distinguished lifting problems in simplicial sets. GT uses them as a design template: expose partial local structure, predict or aggregate the missing component, and test compatibility with the observed faces. This analogy does not identify gradient descent with a horn filler, and universal approximation of functions does not establish a simplicial lifting property.

We first describe more complex ways of extracting parts of categorical structures using simplicial subsets and horns. These structures will play a key role in defining suitable lifting problems.

Definition 37. The **standard simplex** Δ^n is the simplicial set defined by the construction

$$([m] \in \Delta) \mapsto \mathbf{Hom}_{\Delta}([m], [n])$$

By convention, $\Delta^{-1} := \emptyset$. The standard 0-simplex Δ^0 maps each $[n] \in \Delta^{op}$ to the single element set $\{\bullet\}$.

Definition 38. Let $S_{\bullet}$ denote a simplicial set. If for every integer $n \geq 0$, we are given a subset $T_n \subseteq S_n$, such that the face and degeneracy maps

$$d_i : S_n \rightarrow S_{n-1} \quad s_i : S_n \rightarrow S_{n+1}$$

applied to T_n result in

$$d_i : T_n \rightarrow T_{n-1} \quad s_i : T_n \rightarrow T_{n+1}$$

then the collection $\{T_n\}_{n \geq 0}$ defines a **simplicial subset** $T_{\bullet} \subseteq S_{\bullet}$.

Definition 39. The **boundary** is a simplicial set $(\partial \Delta^n) : \Delta^{op} \rightarrow \mathbf{Set}$ defined as

$$(\partial \Delta^n)([m]) = \{\alpha \in \mathbf{Hom}_{\Delta}([m], [n]) : \alpha \text{ is not surjective}\}$$

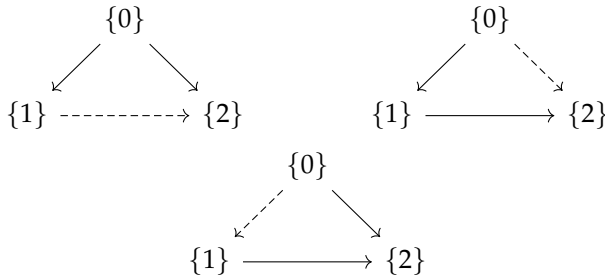
Note that the boundary $\partial\Delta^n$ is a simplicial subset of the standard n -simplex Δ^n .

Definition 40. The **Horn** $\Lambda_i^n : \Delta^{op} \rightarrow \mathbf{Set}$ is defined as

$$(\Lambda_i^n)([m]) = \{\alpha \in \mathbf{Hom}_\Delta([m], [n]) : [n] \not\subseteq \alpha([m]) \cup \{i\}\}$$

Intuitively, the Horn Λ_i^n can be viewed as the simplicial subset that results from removing the interior of the n -simplex Δ^n together with the face opposite its i th vertex.

Consider composing 1-simplices to form a 2-simplex. For each $0 \leq k \leq n$, the horn Λ_k^n is the union of all faces of Δ^n except the k th face. The dashed arrow below indicates the edge missing from each 2-horn.



In the nerve of an ordinary category, every inner 2-horn has a unique filler, namely composition. Outer horns need not have fillers; their existence is related to invertibility. More generally, Kan complexes require fillers for all horns, whereas quasicategories require fillers for inner horns. GT does not claim either property. It uses selected low-dimensional completion tasks as trainable objectives and must report exactly which horns, probes, and residuals are implemented.

Language Modeling With Geometric Transformers

To test whether a GT block can serve as a viable replacement for a standard Transformer block, we use language modeling as a controlled case study. For a token sequence, we instantiate the local template on its 1D simplicial complex underlying a token sequence. Here:

- token positions $i = 1, \dots, L$ are the 0-simplices;
- edges $(i, i + 1)$ form the 1-skeleton (the local adjacency relation);
- the global operator is standard multi-head self-attention over $\{1, \dots, L\}$;
- the local geometric operator is implemented as a depthwise 1D convolution over the sequence, which performs learnable message-passing over the neighbors $\{i - 1, i, i + 1\}$; and
- the pointwise update is provided by a feed-forward network.

Experimental Results on WikiText-103

The decisive experimental variable is not merely architecture but *information regime*. In strict-causal next-token prediction, every feature used at position i must be computable from the prefix through i . An augmented-context geometric branch may read a future token or a symmetric neighborhood containing it. Models from different regimes do not solve the same prediction problem and their perplexities cannot support an architecture-only comparison.

We compare three architectural templates:

- **Transformer**: a standard decoder-only Transformer with L layers and model dimension d .
- **GeomTrans-Lite**: the same Transformer backbone augmented with a local geometric convolutional block (our “GT-Lite” layer) in each block.
- **GeomTrans-Full**: the full block with a simplicial lifting–mixing–readout path and optional registered residuals.

WHAT THE FIRST EXPERIMENT REVEALED. The initial WikiText experiments compared a causal Transformer with GT variants whose local mixer could access future-informed features. Perplexities near 1 were therefore a diagnostic warning, not a headline improvement. A later regime-controlled run at depth 2 and width 256 produced the following WikiText–103 snapshot:

Model	Information regime	Validation PPL	Test PPL
Transformer	strict causal	234.666	232.674
GT-Lite	strict causal	220.219	215.384
Transformer + future hint	augmented	2.728	2.825
GT-Lite + future context	augmented	1.215	1.267

These values establish feasibility and motivate a closer study; they do not show that GT universally outperforms Transformers. A valid comparison also requires matched parameter counts, optimization budgets, repeated seeds, and confidence intervals. The purpose of this chapter is architectural: it identifies the local structural mechanism and the leakage failure that the later information-regime chapter must control.

Constructing Large Causal Models

This second case study asks whether the same local operator can act on a relational corpus rather than a token sequence. A structured ontology contains ten coarse domains (*Economics and Finance*, *Medicine and Healthcare*, *Marketing and Product*, etc.) and one hundred fine-grained subtopics. For the full corpus, prompts generate 1,000 candidate statements for each subtopic,

Table 2: WikiText–103 architecture comparisons must be made within an information regime. The strict-causal GT-Lite gain is modest; the dramatic cross-regime gap is caused primarily by access to additional information. Chapter 9 reports the full causal-ablation suite and protocol.

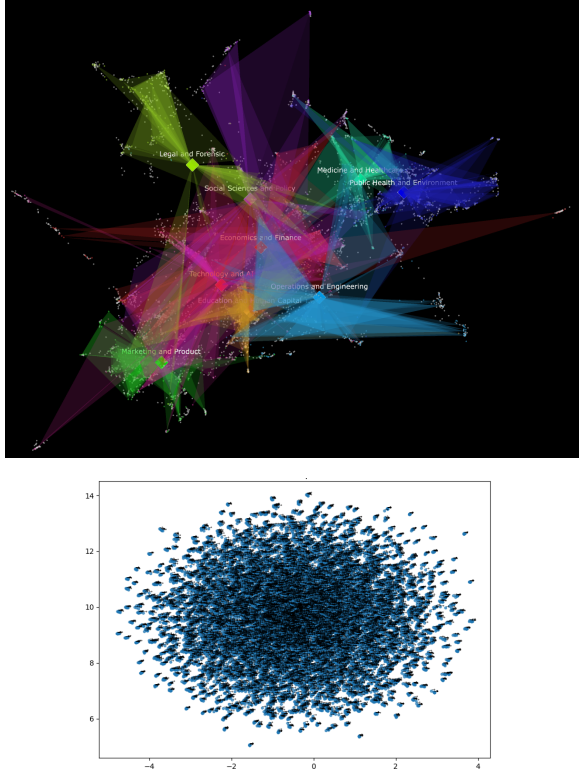


Figure 18: Exploratory UMAP views of the causal-claim embeddings. Top: embeddings after three fixed random geometric message-passing iterations. Bottom: the original normalized SentenceTransformer embeddings. UMAP is a stochastic visualization, so visual separation alone is not evidence of a better causal representation.

yielding 100,000 sentences before deduplication. The final dataset contains 90,016 statements. Each extracted triple is associated with the coarse domain from the corpus metadata. Across the 90K dataset we obtain 54,514 unique entities and 57,390 relational edges. We embed each unique entity using the all-MiniLM-L6-v2 SentenceTransformer, yielding normalized vectors in $\mathbb{R}^{384}$. No fine-tuning is performed. Cosine similarity is used throughout. For each domain, we form 2-simplices (triangles) by connecting all triples (h, r, t_i) and (h, r, t_j) that share the same head entity h . This produces domain-local simplicial patches reflecting the latent relational neighborhood.

Geometric Refinement via DB and GT

We apply three iterations of diagrammatic message passing. In this exploratory run the weights are random and fixed: no supervised objective, DB loss, or gradient update is used. The operation therefore defines a random relational smoothing baseline, not a trained GT result. The two UMAP plots show that the operation changes neighborhood geometry, but they cannot establish that the change preserves causal meaning or improves a downstream task. Such a claim would require quantitative neighborhood-preservation measures, task labels or held-out queries, multiple UMAP seeds, and comparisons with ordinary graph smoothing and random projections. The example is retained to demonstrate that the architectural interface transfers from se-

Domain	# Triangles
Economics and Finance	1140
Education and Human Capital	950
Legal and Forensic	553
Marketing and Product	1059
Medicine and Healthcare	932
Operations and Engineering	1281
Public Health and Environment	1336
Social Sciences and Policy	1015
Technology and AI	782

Table 3: Simplicial complex constructed from a 100K causal claims dataset.

quences to relational complexes; it is not evidence that a causal manifold has been recovered.

A Global–Local Operator Perspective

A standard Transformer layer implements a *global* operator on this complex: self-attention defines a data-dependent kernel $K_{\text{attn}}(i, j; x)$ over all positions $i, j \in \{1, \dots, L\}$, and applies a global mixing step of the form

$$h_i^{\text{attn}} \approx \sum_{j=1}^L K_{\text{attn}}(i, j; x) h_j,$$

followed by a pointwise nonlinearity in the FFN. In contrast, our depth-wise Conv1d layer realizes a purely *local* geometric operator K_{geo} on the 1-skeleton of X , mixing only a fixed neighborhood $\mathcal{N}(i)$ (e.g. $\{i-1, i, i+1\}$).

At a high level,

$$h_i^{\text{geo}} \approx \sum_{j \in \mathcal{N}(i)} K_{\text{geo}}(i, j) h_j,$$

where K_{geo} is a learned local kernel. It becomes a discretized Laplacian or diffusion operator only if symmetry, sign, normalization, and conservation conditions appropriate to such an operator are imposed. GT layers combine this local path with global attention in a single block.

Summary

This chapter introduced GT as a global–local neural architecture whose local features are indexed by a declared finite relational complex. Simplicial sets, horns, and Yoneda supply design language and exact comparison targets; the implemented network realizes only a finite selection of that structure. DB can penalize registered path equations, but neither the architecture nor a low residual proves functoriality, a Kan lifting property, or a universal construction.

The empirical lesson is equally important. Early near-unit perplexities were caused by an augmented information pathway. Under a matched strict-causal regime, GT-Lite showed a modest improvement in the reported WikiText-103 snapshot, while the relational-manifold study remains an exploratory visualization. Chapter 8 next introduces measurable order-sensitivity of learned sub-operators; Chapter 9 then performs the information-regime audit needed to interpret language-model results.

Suggested Reading

The standard Transformer architecture originates with [Vaswani et al. \[2017\]](#). For the categorical and higher-order setting of Geometric Transformers, see [Mahadevan \[2024a\]](#). The topological background is supplied by [Edelsbrunner et al. \[2002\]](#) and [Ghrist \[2014\]](#). For nerves, horns, Kan complexes, and the distinction between categories and general simplicial sets, see [Riehl \[2017\]](#).

Dynamic Compositionality

Every neural network is compositional at the level of its typed forward graph: its modules are functions and the architecture specifies their order. This chapter studies a different, optional property. For two type-compatible sub-operators T_1, T_2 , we compare the counterfactual orders $T_1 \circ T_2$ and $T_2 \circ T_1$ on a declared distribution of states. Their normalized squared discrepancy is called *commutator energy*. It is a forward diagnostic of order sensitivity, not a theorem about gradient descent, Čech cohomology, curvature, or generalization.

The chapter connects this probe to the implementation of Diagrammatic Backpropagation (DB) and Geometric Transformers (GT). Local smoothing, geometric message passing, and mixture routing alter the probe, sometimes substantially. Whether that alteration is beneficial must be established by matched task metrics, information regimes, and ablations. Our goal is a code-faithful account that makes those evidential boundaries explicit.

Probe, Scope, and Falsifiability

Modern neural architectures assemble attention, feed-forward maps, normalization, and residual connections into a typed computation graph. The graph itself already composes. The question here is whether selected modules are sensitive to an artificial reversal of order on states actually visited by the model. Strong order sensitivity may be informative, benign, or necessary; its interpretation depends on the modules and task.

In the experiments reported here, several GT variants have lower commutator energy than the corresponding baseline. We call some aggregated residuals *Čech-style* because they compare maps on declared overlaps. They are not Čech cocycles or cohomology classes unless a cover, coefficient object, coboundary operator, and cocycle condition are explicitly constructed [Bott and Tu, 1982, Ghrist, 2014].

This chapter examines DB and GT at the level of their implemented operators. The aim is to define the following measurement precisely, report where it differs across architectures, and identify—without presupposing—the mechanisms that might drive those differences:

$$\text{Comm}(T_1, T_2; x) = \|T_1(T_2(x)) - T_2(T_1(x))\|_2^2,$$

where T_1, T_2 are concrete, type-compatible sub-operators evaluated on representations x encountered during training. The quantity directly measures only output order sensitivity. Any relationship to parameter-gradient conflict must be measured independently, for example by gradient cosine similarity or an intervention on the auxiliary penalty.

Definition 41. Fix type-compatible maps $T_1, T_2 : X \rightarrow X$, a state distribution μ , a nonnegative normalization $n(x)$, and a tolerance τ . The pair is (μ, τ) -**order-compatible** when

$$\mathbb{E}_{x \sim \mu} \frac{\|T_1(T_2(x)) - T_2(T_1(x))\|_2^2}{n(x)} \leq \tau.$$

Dynamic compositionality is the modeling hypothesis that selected module pairs should remain order-compatible for specified observers during a declared phase of training. It is not a property of composition in general.

COMMUTATOR ENERGY quantifies the discrepancy between two counterfactual compositions. It is meaningful only when both orders are typed and when the state distribution and normalization are reported. A low value is not automatically desirable: identical, constant, or severely contracted operators can commute while being useless.

Čech-style obstruction proxy

Performance curves alone do not explain differences among architectures. We therefore record a computable overlap proxy. Treat an encoder block as a collection of endomaps on the shared residual stream—attention A , feed-forward F , and, in GT variants, a geometric operator C or G . We measure pairwise order sensitivity using

$$\text{Comm}(i, j; x) = \frac{\|T_i(T_j(x)) - T_j(T_i(x))\|^2}{\|x\|^2 + \varepsilon}.$$

The *Čech-style overlap proxy* for a layer is the average commutator energy over a registered list of pairs; the model-level value averages over layers.

Concretely, we use:

- Transformer: average of $\text{Comm}(A, F; x)$,
- GT-Lite: average of $\text{Comm}(A, C; x), \text{Comm}(C, F; x), \text{Comm}(A, F; x)$,
- GT-Full: average of $\text{Comm}(A, F; x), \text{Comm}(F, G; x), \text{Comm}(A, G; x)$,

computed in evaluation mode to remove dropout noise. This aggregation is inspired by overlap consistency, but averaging pairwise commutators does not construct a Čech 1-cocycle and does not certify that local maps glue to a global transformation.

A Minimal Demonstration: Residual MLPs on Two Moons

To illustrate that commutator energy is not specific to Transformers or attention-based models, we construct a minimal demonstration using a simple residual multilayer perceptron (MLP) trained on the classic two-moons classification task. This example serves both as a pedagogical illustration and as a smoke test for Diagrammatic Backpropagation (DB) and Geometric Transformer (GT) principles in their simplest form.

SETUP. We consider three small architectures with matched training hyperparameters but different parameterizations: (i) a baseline residual MLP composed of stacked residual blocks, (ii) an MLP augmented with a GT-Lite local mixing module, and (iii) an MLP augmented with a GT-Full message-passing module on a fixed feature graph. All models are trained to perfect accuracy on the two-moons task within a few epochs.

COMMUTATOR-ENERGY PROBE. During training, we measure the normalized commutator energy between adjacent modules in the residual stack:

$$\text{Comm}(T_i, T_{i+1}; x) = \frac{\|T_i(T_{i+1}(x)) - T_{i+1}(T_i(x))\|_2^2}{\|x\|_2^2 + \varepsilon},$$

averaged over module pairs and minibatches. This quantity captures order sensitivity between learned sub-operators and is computed purely in the forward pass, independent of task loss.

RESULTS. The figure below shows the evolution of commutator energy in the displayed run. All three models reach the same reported task accuracy, while their probe trajectories differ: the baseline rises, GT-Lite is lower, and GT-Full is lowest. Because the architectures themselves change the operators being compared, this is a smoke test rather than an isolated causal effect of the probe penalty.

INTERPRETATION. The example demonstrates that order sensitivity can be measured outside attention models. It does not establish that rising energy is a general failure mode of gradient learning or that lower energy causes stability. A stronger test would hold architecture fixed, vary only the auxiliary penalty, repeat seeds, measure parameter-gradient conflict independently, and include contracted or constant-map controls that achieve low energy for the wrong reason.

Results: obstruction trajectories explain stability

The next figure plots the overlap proxy for one WikiText-103 experiment. The baseline rises, GT-Lite falls and then drifts upward, and GT-Full remains lower. These are descriptive trajectories. The run belongs to the earlier experimental suite in which information pathways were not yet fully

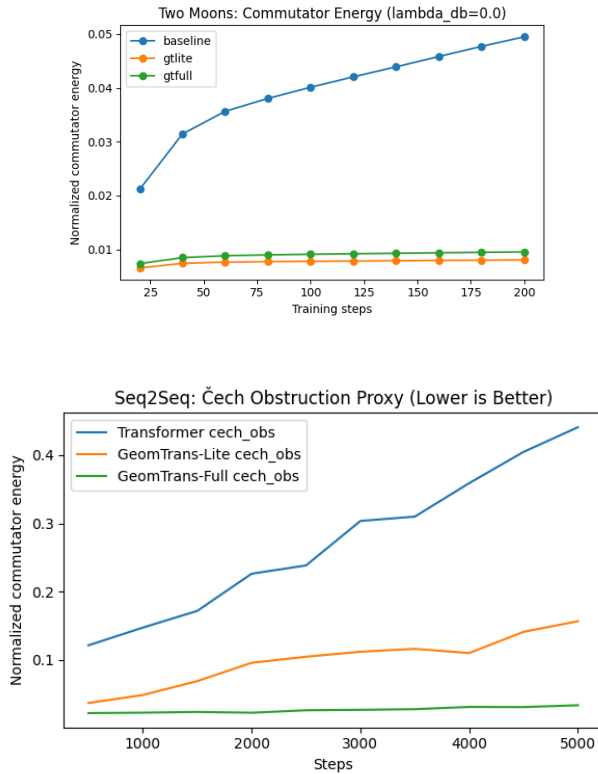


Figure 19: Commutator energy during training on the two-moons task. All models reach the same reported accuracy in this illustrative run, while the GT variants have lower probe values. Since architecture and probe value change together, the plot does not show that low commutator energy causes better optimization or generalization.

Figure 20: **Āech-style overlap proxy during training on WikiText-103.** Normalized commutator energy for the baseline Transformer, GT-Lite, and GT-Full in an early experiment. Lower means less order sensitivity for the registered pairs, not necessarily a better model. The plot is descriptive and is confounded by the information-regime issue analyzed in Chapter 9.

matched; it therefore cannot show that the proxy explains validation behavior or that GT-Full enforces descent. Chapter 9 supplies the required information-regime audit. Here the plot motivates a mechanistic question: which code paths change the measured counterfactual order sensitivity?

Dynamic Compositionality and Order Sensitivity in Residual Learning

Deep networks are *statically compositional*: their forward computation is a composition of modules arranged in a directed acyclic graph. For example, a Transformer block composes attention, residual additions, normalization, and a feed-forward subnetwork. Static compositionality is guaranteed by architecture. In contrast, this chapter uses *dynamic compositionality* for a declared order-compatibility objective on selected learned sub-operators. Even when the forward graph is compositional, two endomaps may be strongly order-sensitive on the representations encountered during training. DB can measure or penalize that selected residual. The phrase does not imply that gradient descent itself fails to compose.

From static composition to learned deformations

Residual architectures provide a natural entry point because each sublayer is explicitly an identity map plus a learned correction. Consider two residual modules T_1 and T_2 acting on a representation $x \in \mathbb{R}^d$:

$$T_1(x) = x + f_1(x), \quad T_2(x) = x + f_2(x),$$

where f_1 and f_2 are learnable maps (e.g., an attention transform, a feed-forward network, a convolutional smoother, or a geometric transport map). This form does *not* assume that modules should commute as functions. It only highlights that learning introduces state-dependent *deformations* of the representation space through f_1 and f_2 .

Although the forward computation is unambiguous (the architecture fixes a specific order), learning can be sensitive to the interaction between these deformations: applying T_1 changes the input seen by T_2 , and vice versa. When these interactions are mild, swapping the selected operators changes the output little on the probed states. When they are strong, the counterfactual orders differ. Whether that sensitivity makes training fragile is a separate empirical question.

Order sensitivity and commutator energy

To make this notion of interaction operational, we introduce a forward-only probe that measures *order sensitivity* on representations encountered during training. Given two sub-operators T_1 and T_2 , define the *commutator residual* at x as

$$r(T_1, T_2; x) = T_1(T_2(x)) - T_2(T_1(x)).$$

We then define the corresponding *commutator energy* as the mean-squared magnitude of this residual, optionally normalized by the input energy:

$$E_{\text{comm}}(T_1, T_2; x) = \frac{\|r(T_1, T_2; x)\|_2^2}{\|x\|_2^2 + \varepsilon}.$$

In our implementation, this is computed exactly as:

$$E_{\text{comm}}(T_1, T_2; x) \propto \text{MSE}(T_1(T_2(x)) - T_2(T_1(x))),$$

matching the code fragment `mse (T1 (T2 (x)) - T2 (T1 (x)))`. Importantly, we do *not* interpret small commutator energy as a requirement that T_1 and T_2 commute as abstract functions. Rather, commutator energy is used as an empirical measurement of how strongly the two counterfactual orders differ on the states the model actually visits. Large values show a lack of output exchangeability for those selected maps; they do not directly measure parameter-update interference.

WHY ORDER SENSITIVITY MATTERS FOR LEARNING. E_{comm} is computed from outputs, whereas gradient interference is a relation among parameter gradients. Neither quantity determines the other without additional hypotheses. The testable conjecture is that high order sensitivity sometimes accompanies conflicting optimization signals. It should be evaluated by logging both quantities and by intervening on the commutator penalty while holding architecture and information fixed.

A minimal residual example

To build intuition, consider two residual maps $T_1(x) = x + f_1(x)$ and $T_2(x) = x + f_2(x)$ applied to the same input x . The two compositions expand as

$$\begin{aligned} T_1(T_2(x)) &= x + f_2(x) + f_1(x + f_2(x)), \\ T_2(T_1(x)) &= x + f_1(x) + f_2(x + f_1(x)). \end{aligned}$$

Their difference arises entirely from the *state dependence* of the learned deformations: $f_1(\cdot)$ and $f_2(\cdot)$ are evaluated at different perturbed inputs. When these perturbations strongly change the behavior of the other module, commutator energy is large. When their effects are compatible, commutator energy is small. This provides an intuitive explanation for why commutator energy is a useful descriptive diagnostic even for architectures whose forward graph is fixed.

ROADMAP. We instantiate these definitions inside Transformer encoder blocks by taking T_1, T_2 to be concrete sub-stages such as attention, feed-forward, convolutional smoothing, geometric transport, and mixture-of-experts mixing. We then report how the probe changes across the implemented variants, without treating that change alone as a mechanistic explanation of task performance.

What the Minimal Residual Example Establishes

To isolate the mechanism underlying commutator energy without architectural complexity, we consider a minimal residual multilayer perceptron (MLP) example. The goal is not high performance but calibration: the algebra above predicts the numerical probe exactly. The example shows that order sensitivity can arise in a residual model; it does not identify its optimization consequences.

Two interacting residual blocks

Let $x \in \mathbb{R}^d$ be an input representation and consider two residual modules applied sequentially:

$$T_1(x) = x + f_1(x), \quad T_2(x) = x + f_2(x),$$

where f_1 and f_2 are small MLPs with independent parameters. The composite forward computation is fixed by architecture, for example $T_2(T_1(x))$. Statically, this model is compositional: it is simply a composition of two functions.

During training, f_1 and f_2 are updated from a shared loss and remain state-dependent. Their output commutator and their parameter-gradient interaction are distinct observables.

Order sensitivity in residual updates

Although the architecture applies T_1 before T_2 , we can probe the learning dynamics by comparing the two compositions

$$T_2(T_1(x)) \quad \text{and} \quad T_1(T_2(x)).$$

Expanding these expressions reveals the source of order sensitivity:

$$\begin{aligned} T_2(T_1(x)) &= x + f_1(x) + f_2(x + f_1(x)), \\ T_1(T_2(x)) &= x + f_2(x) + f_1(x + f_2(x)). \end{aligned}$$

The difference between these two outputs arises entirely from the *state dependence* of f_1 and f_2 . When the perturbation induced by one module significantly alters the input seen by the other, the two compositions differ substantially. This discrepancy is precisely what the commutator-energy probe measures:

$$E_{\text{comm}}(T_1, T_2; x) = \|T_2(T_1(x)) - T_1(T_2(x))\|_2^2.$$

Importantly, large commutator energy does not indicate a violation of any architectural constraint. Rather, it indicates that the two learned deformations interact in an order-sensitive way on the states encountered during training.

A conjectured connection to gradient interference

It is plausible that order-sensitive maps sometimes produce conflicting parameter gradients, but the commutator does not contain those gradients. A calibrated experiment should compare it with, for example, cosine similarity between module-specific gradients and test whether penalizing one changes the other. Until that experiment is performed, gradient interference remains a hypothesis rather than the meaning of the probe.

Implications for Diagrammatic Backpropagation

Diagrammatic Backpropagation can add an auxiliary objective that discourages large order sensitivity. Minimizing that objective does push f_1 and f_2 toward commuting on the sampled states, but it can also favor contraction or loss of expressivity. Task loss, variance and rank diagnostics, and matched ablations are therefore part of the admission test.

Transformer blocks replace f_1 and f_2 with attention, feed-forward, and geometric operators. The same residual can be computed there, although its semantic and optimization significance must be re-established at that scale.

Dynamic Compositionality in Transformer and Geometric Transformer Blocks

Previously we introduced the notion of dynamic compositionality as an order-compatibility objective. We now instantiate the probe inside Transformer encoder blocks and their geometric variants.

Transformer encoder blocks as interacting sub-operators

A standard Transformer encoder block consists of a small number of sub-stages applied sequentially to the same representation: multi-head self-attention, residual addition, layer normalization, and a positionwise feed-forward network. Although the execution order is fixed by the architecture, learning dynamics depend on how these sub-stages interact on the representation manifold visited during training.

For the purposes of dynamic compositionality analysis, we group each sub-stage together with its residual connection and normalization into a single effective sub-operator. In the baseline Transformer, we therefore identify two concrete sub-operators:

- A : the attention stage, including residual addition and layer normalization;
- F : the feed-forward stage, including residual addition and layer normalization.

These are exactly the operators implemented in the probe as: $A(z) = \text{LN}(z + \text{Attn}(z))$ and $F(z) = \text{LN}(z + \text{FF}(z))$.

Measuring order sensitivity inside a block

Although the Transformer block applies A before F in the forward pass, the learning dynamics depend on how strongly these sub-operators perturb each other’s effective inputs. To probe this interaction, we evaluate the commutator-energy diagnostic

$$E_{\text{comm}}(A, F; x) = \frac{\|A(F(x)) - F(A(x))\|_2^2}{\|x\|_2^2 + \epsilon},$$

computed on the actual block input x encountered during training. This quantity measures how sensitive the learned representation update is to the order in which attention and feed-forward transformations are applied. Importantly, we do not require A and F to commute as functions. Large commutator energy means only that this counterfactual reordering changes the block output on the sampled states.

Why vanilla Transformers exhibit high commutator energy

In a baseline Transformer, self-attention introduces global, token-coupling effects that depend sharply on the current representation, while the feed-forward network applies nonlinear, positionwise transformations. Layer normalization and residual connections tightly couple these effects across feature dimensions. As a result, applying attention may alter the input seen by the feed-forward network substantially, and reversing the operators can produce high commutator energy. Whether that value correlates with instability or poor generalization is dataset- and regime-dependent and must be reported rather than inferred from architecture.

Geometric Transformers

We introduced a family of GT models in the previous chapter, which we now analyze from the standpoint of dynamic compositionality.

GT-Lite: reducing order sensitivity via local smoothing

GT-Lite augments the Transformer block with an additional sub-operator C that performs local geometric smoothing, implemented as a convolution or local neighborhood mixer. In this case, the probe measures the average of pairwise order sensitivities:

$$E_{\text{comm}}(A, C; x), \quad E_{\text{comm}}(C, F; x), \quad E_{\text{comm}}(A, F; x).$$

A convolution can reduce high-frequency variation in the sequence index, but a learned kernel is not necessarily smoothing. The hypothesis is that the implemented local branch regularizes states enough to reduce selected commutators. Kernel spectra and matched ablations are needed to distinguish that mechanism from normalization, residual scaling, or additional capacity.

GT-Full: geometric transport and alignment of representation geometry

The full Geometric Transformer replaces local smoothing with an explicit geometric transport operator G , implemented via simplicial lifting, message passing, and readout over a token-interaction complex. The corresponding probe measures:

$$E_{\text{comm}}(A, G; x), \quad E_{\text{comm}}(F, G; x), \quad E_{\text{comm}}(A, F; x).$$

The implemented operator propagates messages along selected edges and reads them back to token states. Calling this *coordinate alignment* is a geometric interpretation, not a property established by the code. Lower commutator values are consistent with that interpretation, but alternatives include contraction, normalization, residual scaling, and changed information flow. Representation rank and variance controls are needed before claiming that expressivity is preserved.

GT-MoE: routing-induced order sensitivity

In GT-MoE, the feed-forward stage is replaced by a mixture-of-experts operator M whose effective transformation depends on gating decisions computed from the current representation. The probe measures $E_{\text{comm}}(A, M; x)$. Routing introduces state-dependent branching, so its interaction with attention can be order-sensitive. This is a candidate explanation for observed probe differences, not a deduction from mixture routing alone.

Summary: architectural control of dynamic compositionality

Across all variants, commutator energy supplies one consistently computed diagnostic. The architectural comparisons generate hypotheses about local smoothing, message passing, and routing. They do not by themselves identify the cause of task-performance differences.

From Mechanism to Implementation: Explaining the DB+GT Code Path

Sections III–III defined commutator energy as a forward diagnostic of order sensitivity. We now explain how it is implemented for Transformer and GT models. The code determines the measured quantity; it does not make task-level behavior follow automatically. Our goal in this section is to demystify the DB+GT implementation by mapping each conceptual component to the corresponding computational step.

The commutator-energy primitive

At the lowest level, the DB instrumentation relies on a single primitive:

$$\text{Comm}(T_1, T_2; x) = \|T_1(T_2(x)) - T_2(T_1(x))\|_2^2,$$

implemented directly as:

```
_mse (T1 (T2 (x)) - T2 (T1 (x)))
```

This computation introduces no parameters and can be evaluated without gradients. Division by $\|x\|_2^2 + \epsilon$ reduces sensitivity to input magnitude, but does not make the statistic scale-invariant for nonlinear maps or automatically comparable across architectures. Operator definitions, masks, evaluation mode, precision, and the sampling distribution must be held fixed or reported.

Crucially, this primitive is agnostic to the nature of T_1 and T_2 ; they are concrete callable subroutines corresponding to actual sub-stages of a model (e.g., attention, feed-forward, convolutional smoothing, geometric transport, or mixture-of-experts routing).

Layerwise instrumentation in Transformer blocks

The function `encoder_cech_obstruction` evaluates commutator energy inside each encoder block by extracting its internal sub-operators. For a baseline Transformer block, these are:

- A : the self-attention stage together with residual addition and layer normalization;
- F : the feed-forward stage together with residual addition and layer normalization.

The instrumentation computes

$$E_{\text{comm}}(A, F; x_{\text{in}})$$

on the block input x_{in} and then advances the forward pass normally. The resulting scalar measures how strongly the two output orders differ on the representations encountered at that layer.

Because this computation is performed under `@torch.no_grad()`, it does not affect training unless explicitly included as a loss term. Instead, it provides a per-layer diagnostic of dynamic compositionality.

Extending the probe to Geometric Transformer variants

Geometric Transformer variants introduce additional sub-operators, and the instrumentation generalizes by measuring pairwise order sensitivity among them.

GT-LITE. GT-Lite augments the block with a local geometric smoothing operator C (e.g., a 1D convolution over neighboring tokens). The probe evaluates the average of:

$$E_{\text{comm}}(A, C; x), \quad E_{\text{comm}}(C, F; x), \quad E_{\text{comm}}(A, F; x),$$

capturing pairwise order sensitivity among attention, the local branch, and the feed-forward update. The reported runs have lower values, but the probe does not identify smoothing as the cause.

GT-FULL. The full Geometric Transformer introduces an explicit geometric transport operator G implemented via simplicial lifting, message passing, and readout. The probe evaluates:

$$E_{\text{comm}}(A, G; x), \quad E_{\text{comm}}(F, G; x), \quad E_{\text{comm}}(A, F; x).$$

The message-passing branch is interpreted as transport across tokens. Lower reported energy is consistent with coordinate alignment, but neither curvature reduction nor preservation of expressivity follows from this statistic alone.

GT-MoE. In GT-MoE, the feed-forward stage is replaced by a mixture-of-experts operator M whose effective transformation depends on routing probabilities. The probe measures $E_{\text{comm}}(A, M; x)$, capturing order sensitivity involving state-dependent expert selection. Routing supplies one hypothesis for intermediate and variable probe behavior.

Why commutator energy tracks training stability

Whether commutator energy tracks training stability is an empirical hypothesis. A correlation can be confounded by architecture, normalization, residual scale, representation contraction, or information access. A causal test varies the auxiliary coefficient within a fixed architecture and measures task performance, representation rank, and gradient conflict. DB does not guarantee that lowering the residual preserves expressivity.

From instrumentation to control

In the previous chapter, commutator energy is primarily used as an instrumental probe. When Diagrammatic Backpropagation is enabled, the same quantity (or its generalization to higher-order overlap patterns) is incorporated into the training objective as an auxiliary loss. This transforms a passive diagnostic into an active control mechanism that directly shapes learning dynamics.

From an implementation standpoint, DB adds overlap-residual terms and their gradients to the task objective. Reduced training-set commutator energy is the direct target. Improved stability, generalization, and global structure do not follow directly and require independent evaluation.

SUMMARY. This section has isolated the code-level measurement: evaluate two typed orders, normalize their output discrepancy, and optionally add it to the loss. The choice of pairs and the claim that their order should matter little remain modeling decisions to validate for each task.

GT-Lite Under the Hood: Transformer Blocks with Local Geometric Smoothing

We now unpack the simplest Geometric Transformer variant, *GT-Lite*, at the level of concrete PyTorch code. The purpose of this section is to explain how a local sequence operator is inserted between attention and feed-forward stages and how the probe is evaluated around it. This section is intended to serve as a code-level reference for readers wishing to understand or extend the DB+GT implementation.

GeomEncoderBlock: structure and intent

Listing 1 shows one GT-Lite encoder implementation. Its additional operator is a full-channel one-dimensional convolution (`groups=1`, the PyTorch default), not a depthwise convolution. It has a local receptive field in position but may mix all feature channels. Other experiments in the book use depthwise variants; the two must not be conflated.

Listing 1: GT-Lite encoder block

```
class GeomEncoderBlock(nn.Module):
    def __init__(self, d_model: int, n_heads: int, dim_ff: int, dropout: float = 0.1):
        super().__init__()
        self.self_attn = nn.MultiheadAttention(
            d_model, n_heads, dropout=dropout, batch_first=True
        )
        self.conv = nn.Conv1d(d_model, d_model, kernel_size=3, padding=1)

        self.lin1 = nn.Linear(d_model, dim_ff)
        self.lin2 = nn.Linear(dim_ff, d_model)

        self.ln_attn = nn.LayerNorm(d_model)
        self.ln_conv = nn.LayerNorm(d_model)
        self.ln_ff = nn.LayerNorm(d_model)

        self.dropout = nn.Dropout(dropout)
        self.dropout_ff = nn.Dropout(dropout)

    def forward(self, x: torch.Tensor) -> torch.Tensor:
        attn_out, _ = self.self_attn(x, x, x, need_weights=False)
        x = self.ln_attn(x + self.dropout(attn_out))

        x_conv = self.conv(x.transpose(1, 2)).transpose(1, 2)
        x = self.ln_conv(x + 0.2 * x_conv)

        ff = self.lin2(self.dropout_ff(F.relu(self.lin1(x))))
        x = self.ln_ff(x + ff)
        return x
```

Sub-operators and dynamic compositionality

From the perspective of dynamic compositionality, the GT-Lite encoder block consists of three interacting sub-operators, each applied with a residual connection and layer normalization:

- **Attention operator** A : global token mixing via multi-head self-attention,

$$A(x) = \text{LN}_{\text{attn}}(x + \text{Attn}(x)).$$

- **Local convolution operator** C : convolution over neighboring token positions,

$$C(x) = \text{LN}_{\text{conv}}(x + \alpha \text{Conv}(x)), \quad \alpha = 0.2.$$

- **Feed-forward operator** F : positionwise nonlinear deformation,

$$F(x) = \text{LN}_{\text{ff}}(x + \text{FF}(x)).$$

The overall block computes $F(C(A(x)))$. Architecturally, this is a fixed composition. Dynamically, however, learning depends on how strongly these sub-operators interfere when applied to the same representation.

Why local smoothing reduces commutator energy

The commutator-energy probe evaluates order sensitivity between sub-operators, for example

$$E_{\text{comm}}(A, F; x) = \|A(F(x)) - F(A(x))\|_2^2.$$

In a baseline Transformer, A and F interact directly. Attention introduces sharp, token-dependent changes to the representation, and the feed-forward network is highly sensitive to such changes. As a result, $A(F(x))$ and $F(A(x))$ often differ substantially, producing large commutator energy.

GT-Lite inserts the local operator C . A learned full-channel convolution need not average its neighborhood or act as a low-pass filter; that would require inspecting or constraining its learned frequency response. The reported pairwise commutator energies

$$E_{\text{comm}}(A, C; x), \quad E_{\text{comm}}(C, F; x), \quad E_{\text{comm}}(A, F; x)$$

are lower in the displayed runs. Establishing why requires ablations for kernel width, channel grouping, residual scale, normalization, and parameter count.

GeomDecoderBlock: a non-causal decoder caution

Listing 2 shows the corresponding decoder block. Although self-attention receives a causal mask, the subsequent kernel-size-3 convolution uses symmetric padding. Its output at position i can therefore depend on the representation at $i + 1$. As written, this block is not a valid strict-autoregressive decoder.

For strict autoregressive generation, the convolution must use left-only padding (or an equivalent causal implementation), and all probe evaluations must preserve that mask and shift policy. This small code detail is the source of the information-regime distinction developed in Chapter 9.

Listing 2: GT-Lite decoder block

```

class GeomDecoderBlock(nn.Module):
    def __init__(self, d_model: int, n_heads: int, dim_ff: int, dropout: float = 0.1):
        super().__init__()
        self.self_attn = nn.MultiheadAttention(
            d_model, n_heads, dropout=dropout, batch_first=True
        )
        self.cross_attn = nn.MultiheadAttention(
            d_model, n_heads, dropout=dropout, batch_first=True
        )
        self.conv = nn.Conv1d(d_model, d_model, kernel_size=3, padding=1)

        self.lin1 = nn.Linear(d_model, dim_ff)
        self.lin2 = nn.Linear(dim_ff, d_model)

        self.ln_self = nn.LayerNorm(d_model)
        self.ln_cross = nn.LayerNorm(d_model)
        self.ln_conv = nn.LayerNorm(d_model)
        self.ln_ff = nn.LayerNorm(d_model)

        self.dropout = nn.Dropout(dropout)
        self.dropout_ff = nn.Dropout(dropout)

    def forward(self, x, enc_out, causal_mask):
        self_out, _ = self.self_attn(x, x, x, attn_mask=causal_mask, need_weights=False)
        x = self.ln_self(x + self.dropout(self_out))

        cross_out, _ = self.cross_attn(x, enc_out, enc_out, need_weights=False)
        x = self.ln_cross(x + self.dropout(cross_out))

        x_conv = self.conv(x.transpose(1, 2)).transpose(1, 2)
        x = self.ln_conv(x + 0.2 * x_conv)

        ff = self.lin2(self.dropout_ff(F.relu(self.lin1(x))))
        x = self.ln_ff(x + ff)
        return x

```

SUMMARY. GT-Lite demonstrates how to instrument a local branch with the same order- sensitivity probe. The implementation also demonstrates why architectural claims must include padding, grouping, and information-flow details.

GT-Full Under the Hood: Simplicial Transport as Coordinate Alignment

GT-Full replaces the convolutional GT-Lite branch with an explicit graph-based transport operator over token positions. This section unpacks the block line by line and separates what the code establishes from two empirical hypotheses: that transport lowers selected commutator residuals and that it does so without sacrificing task-relevant global structure.

GeomFullEncoderBlockSeq: code structure

Listing 3 and Listing 4 show the implementation of the GT-Full encoder block. The block retains the standard Transformer attention and feed-forward stages, but inserts an additional geometric transport operator that acts on a graph (or simplicial complex) over token positions.

Sub-operators in GT-Full

From the perspective of dynamic compositionality, the GT-Full encoder block contains three interacting sub-operators:

- **Attention operator** A , defined as

$$A(x) = \text{LN}_{\text{attn}}(x + \text{Attn}(x)).$$

- **Feed-forward operator** F , defined as

$$F(x) = \text{LN}_{\text{ff}}(x + \text{FF}(x)).$$

- **Geometric transport operator** G , defined by lifting token representations into a structured graph space, performing message passing, and reading back into the token space.

The overall block computes $G(F(A(x)))$. As before, the architecture fixes this order; the question addressed by the commutator-energy probe is how sensitive learning dynamics are to alternative local compositions of these operators.

What geometric transport actually does

The geometric transport operator G acts on the flattened representation $h \in \mathbb{R}^{(B \cdot L) \times d}$ and propagates information along a fixed positional graph defined

Listing 3: GT-Full encoder block

```

class GeomFullEncoderBlockSeq(nn.Module):
    def __init__(
        self,
        d_model: int,
        n_heads: int,
        dim_ff: int,
        max_len_src: int,
        num_rel: int = 1,
        dropout: float = 0.1,
        gt_depth: int = 1,
    ):
        super().__init__()
        self.self_attn = nn.MultiheadAttention(
            embed_dim=d_model,
            num_heads=n_heads,
            dropout=dropout,
            batch_first=True,
        )
        self.lin1 = nn.Linear(d_model, dim_ff)
        self.lin2 = nn.Linear(dim_ff, d_model)

        self.ln_attn = nn.LayerNorm(d_model)
        self.ln_ff = nn.LayerNorm(d_model)

        self.dropout_attn = nn.Dropout(dropout)
        self.dropout_ff = nn.Dropout(dropout)

        self.gt = GeometricTransformerV2(
            dim=d_model,
            depth=gt_depth,
            num_rel=num_rel,
        )

        # fixed positional graph over sequence indices
        src_indices, dst_indices = [], []
        for i in range(max_len_src - 1):
            src_indices.extend([i, i + 1])
            dst_indices.extend([i + 1, i])
        edge_index = torch.tensor([src_indices, dst_indices], dtype=torch.long)

        self.register_buffer("edge_index", edge_index)
        self.register_buffer("rel_ids", torch.zeros(edge_index.size(1), dtype=torch.long))
        self.register_buffer("dom_ids", torch.zeros(edge_index.size(1), dtype=torch.long))

```

Listing 4: GT-Full encoder block

```

class GeomFullEncoderBlockSeq(nn.Module):
    def __init__(
        self,
        d_model: int,
        n_heads: int,
        dim_ff: int,
        max_len_src: int,
        num_rel: int = 1,
        dropout: float = 0.1,
        gt_depth: int = 1,
    ):
        ...

    def forward(self, x: torch.Tensor) -> torch.Tensor:
        B, L, D = x.shape

        attn_out, _ = self.self_attn(x, x, x, need_weights=False)
        x = self.ln_attn(x + self.dropout_attn(attn_out))

        ff = self.lin2(self.dropout_ff(F.relu(self.lin1(x))))
        x = self.ln_ff(x + ff)

        # geometric transport
        h = x.reshape(B * L, D)
        edge_index = self.edge_index[:, self.edge_index[0] < L]
        rel_ids = self.rel_ids[self.edge_index[0] < L]
        dom_ids = self.dom_ids[self.edge_index[0] < L]
        h_ref = self.gt(h, edge_index, rel_ids, dom_ids)
        return h_ref.reshape(B, L, D)

```

by `edge_index`. Each node corresponds to a token position, and edges encode adjacency relations. Internally, the `GeometricTransformerV2` module performs multiple rounds of message passing, updating each token’s representation based on its neighbors and relation embeddings.

This process exchanges information along the registered positional graph. In the displayed implementation that graph is a bidirectional chain, so one round of message passing is local and several rounds enlarge the receptive field. Describing the resulting states as sharing a geometric coordinate system is a useful interpretation, but the code alone proves neither coordinate alignment nor global mixing.

Testing the transport hypothesis

The commutator-energy probe for GT-Full evaluates

$$E_{\text{comm}}(A, F; x), \quad E_{\text{comm}}(A, G; x), \quad E_{\text{comm}}(F, G; x).$$

The reported runs associate GT-Full with lower values of these terms. A candidate explanation is that repeated transport makes the states presented to A , F , and G more mutually compatible. The statistic does not identify that mechanism: contraction, normalization, residual scale, capacity, and the bidirectional information path can also lower it. A convincing test therefore compares matched models while ablating the graph, transport depth, directionality, normalization, and residual strength.

The phrase *curvature mismatch* should be reserved for an independently defined differential-geometric quantity. The forward commutator measured here is evidence of order sensitivity, not by itself a curvature estimator.

Expressivity is a separate admission test

Low commutator energy admits trivial solutions, including contraction and representation collapse. Expressivity must therefore be checked independently through task performance, representation rank or variance, and matched probes of long-range dependence. The experiments are consistent with GT-Full retaining more global structure than GT-Lite, but they do not establish that transport is the unique cause or that the property holds across tasks.

SUMMARY. GT-Full supplies a concrete transport operator and a testable architectural hypothesis. In the reported runs it is associated with lower order sensitivity; matched ablations and independent expressivity tests are required before that association can be interpreted causally.

GT-MoE Under the Hood: Routing-Induced Order Sensitivity

The GT-MoE variant replaces the standard feed-forward sublayer with a mixture-of-experts (MoE) module whose effective transformation depends on learned routing decisions. While GT-MoE often achieves strong task-level performance and reduces commutator energy relative to a baseline Transformer, its mechanism differs fundamentally from that of GT-Full. In this section, we unpack the GT-MoE encoder block at the level of concrete code and explain how state-dependent routing alters dynamic compositionality.

GeomEncoderMoEBlock: code structure

Listing 5 shows the implementation of the GT-MoE encoder block. The block retains standard self-attention and residual structure, but replaces the feed-forward network with a gated mixture of experts.

Sub-operators in GT-MoE

From the perspective of dynamic compositionality, the GT-MoE block contains two primary sub-operators:

- **Attention operator** A , defined as

$$A(x) = \text{LN}_{\text{attn}}(x + \text{Attn}(x)).$$

- **Mixture-of-experts operator** M , defined as

$$M(x) = \text{LN}_{\text{ff}}\left(x + \sum_{e=1}^E g_e(x) E_e(x)\right),$$

where $g_e(x)$ are learned gating probabilities and E_e are expert feed-forward networks.

The overall block computes $M(A(x))$. Unlike the standard feed-forward network, the effective transformation implemented by M depends explicitly on routing decisions computed from the current representation.

Routing as state-dependent branching

The defining feature of GT-MoE is that routing decisions introduce conditional computation. For two nearby representations x and x' , the set of active experts and their relative weights can differ substantially. As a result, the effective deformation of the representation space induced by M can change discontinuously as a function of x .

From the perspective of dynamic compositionality, this introduces an additional source of order sensitivity: applying attention before routing can change which experts are selected, while applying routing before attention

Listing 5: GT-MoE encoder block

```

class GeomEncoderMoEBlock(nn.Module):
    def __init__(
        self,
        d_model: int,
        n_heads: int,
        dim_ff: int,
        n_experts: int = 4,
        dropout: float = 0.1,
        top_k: int = 2,
    ):
        super().__init__()
        self.self_attn = nn.MultiheadAttention(
            d_model, n_heads, dropout=dropout, batch_first=True
        )

        self.experts = nn.ModuleList([
            nn.Sequential(
                nn.Linear(d_model, dim_ff),
                nn.GELU(),
                nn.Linear(dim_ff, d_model),
            )
            for _ in range(n_experts)
        ])

        self.gate = nn.Linear(d_model, n_experts)

        self.ln_attn = nn.LayerNorm(d_model)
        self.ln_ff = nn.LayerNorm(d_model)
        self.dropout = nn.Dropout(dropout)

    def forward(self, x: torch.Tensor) -> torch.Tensor:
        attn_out, _ = self.self_attn(x, x, x, need_weights=False)
        x = self.ln_attn(x + self.dropout(attn_out))

        gate_logits = self.gate(x)
        gate_probs = gate_logits.softmax(dim=-1)

        if self.top_k < gate_probs.size(-1):
            top_vals, top_idx = gate_probs.topk(self.top_k, dim=-1)
            sparse = torch.zeros_like(gate_probs)
            sparse.scatter_(2, top_idx, top_vals)
            gate_probs = sparse / (sparse.sum(dim=-1, keepdim=True) + 1e-8)

        ff_out = torch.zeros_like(x)
        for e_idx, expert in enumerate(self.experts):
            w_e = gate_probs[..., e_idx].unsqueeze(-1)
            ff_out = ff_out + w_e * expert(x)

        x = self.ln_ff(x + self.dropout(ff_out))

```

would alter the attention input in a different way. The commutator-energy probe captures this effect by measuring

$$E_{\text{comm}}(A, M; x) = \|A(M(x)) - M(A(x))\|_2^2.$$

A routing hypothesis for intermediate energy

In the reported runs, GT-MoE occupies an intermediate commutator-energy regime. One plausible explanation is that expert specialization separates some transformations while state-dependent routing introduces new order sensitivity, especially near top- k boundaries. This explanation does not follow from the architecture alone. It can be tested by comparing dense and sparse gates, varying gate temperature and expert count, measuring routing flips under small perturbations, and matching parameter count to non-MoE controls.

Comparison with GT-Lite and GT-Full

The three blocks motivate distinct candidate mechanisms:

- GT-Lite adds local convolutional mixing.
- GT-Full adds graph-indexed transport.
- GT-MoE adds state-dependent expert routing.

Their effects on order sensitivity, expressivity, optimization, and information flow must be measured separately rather than read off from these labels.

SUMMARY. The code makes routing a distinct, inspectable source of state dependence. The intermediate empirical behavior is a result to explain, not yet a proof that routing reduces interference or that transport plays a unique causal role.

Instrumenting and Controlling Order Sensitivity

The following pseudocode distinguishes two operations that must not be collapsed. Algorithm 6 is a diagnostic evaluated in `eval` mode under `no_grad`. If the same residual is used as an auxiliary training objective, it must be recomputed by a differentiable routine with gradients enabled; a value returned by the diagnostic probe cannot train the model. Diagrammatic constraint losses are a third object and require an explicit diagram instance rather than merely a pair of neural sub-operators.

Algorithm 7 gives the corrected conceptual training loop. Every sub-operator evaluation must receive the same registered attention mask, padding policy, positional encoding, and training/evaluation state as the path it is intended to compare.

Algorithm 8 gives the intended batched transport operator. A crucial implementation detail is absent from the displayed `GeomFullEncoderBlockSeq.forward`: after flattening to shape $(B \cdot L, d)$, the length- L graph must be restricted on *both* endpoints, replicated for every batch element, and offset by bL . Filtering only the source endpoint can retain an edge into node L ; without batch offsets, the listed indices address only the first sample (and can cross a batch boundary). The corrected pseudocode makes both operations explicit. The positional graph used here is a bidirectional chain; richer typed graphs require their own information-flow audit.

Algorithm 9 defines the convolutional operator C . The learned kernel mixes adjacent positions, but is not necessarily a low-pass filter. Nor does a numerical choice such as $\alpha = 0.2$ guarantee a small correction, because the convolution norm is learned. The decoder version requires left-only padding for autoregressive use; the symmetric padding shown earlier accesses a future position. These properties must be measured or enforced rather than inferred from the name “smoothing.”

Finally, Algorithm 10 defines the GT-MoE mixing operator M . Algorithm 10 corresponds to the MoE stage in `GeomEncoderMoEBlock`. The gating distribution p is computed per token and optionally sparsified via top- k routing. Expert outputs are then aggregated as a weighted sum. Because routing is state-dependent, small changes to z can change which experts dominate, introducing a distinct form of order sensitivity relative to GT-Lite and GT-Full.

Algorithm 6: Layerwise Commutator-Energy Diagnostic

Input: Model M with encoder layers $\{L_\ell\}_{\ell=1}^L$; input tokens $s \in \mathbb{N}^{B \times T}$;
small constant $\varepsilon > 0$

Output: Scalar obstruction score $\text{Obs}(M; s)$ and per-layer scores $\{\text{Obs}_\ell\}_{\ell=1}^L$

- 1: **Set** M to eval mode; save previous training flag.
- 2: $x \leftarrow \text{PosEnc}(\text{TokEmb}(s))$ $\triangleright x \in \mathbb{R}^{B \times T \times d}$
- 3: $\text{ObsList} \leftarrow []$
- 4: **for** $\ell = 1$ **to** L **do**
- 5: $x_{\text{in}} \leftarrow x$
- 6: $\text{denom} \leftarrow \text{MSE}(x_{\text{in}}) + \varepsilon$ $\triangleright$ scale normalization
- 7: **// Define sub-operators as in the actual forward block (including residual & LN).**
- 8: **if** L_ℓ is a baseline Transformer encoder block **then**
- 9: $A(z) \leftarrow \text{LN}_1(z + \text{Drop}(\text{Attn}(z)))$
- 10: $F(z) \leftarrow \text{LN}_2(z + \text{FF}(z))$
- 11: $\text{Obs}_\ell \leftarrow \frac{\text{MSE}(A(F(x_{\text{in}})) - F(A(x_{\text{in}})))}{\text{denom}}$
- 12: $x \leftarrow L_\ell(x)$
- 13: **else if** L_ℓ is GT-Lite (attention + conv smoothing + FF) **then**
- 14: $A(z) \leftarrow \text{LN}_{\text{attn}}(z + \text{Drop}(\text{Attn}(z)))$
- 15: $C(z) \leftarrow \text{LN}_{\text{conv}}(z + \alpha \cdot \text{Conv}(z))$ $\triangleright \alpha \approx 0.2$
- 16: $F(z) \leftarrow \text{LN}_{\text{ff}}(z + \text{FF}(z))$
- 17: $\text{Obs}_\ell \leftarrow \frac{1}{3 \text{denom}} \left(\text{MSE}(A(C(x_{\text{in}})) - C(A(x_{\text{in}}))) + \right.$
 $\left. \text{MSE}(C(F(x_{\text{in}})) - F(C(x_{\text{in}}))) + \text{MSE}(A(F(x_{\text{in}})) - F(A(x_{\text{in}}))) \right)$
- 18: $x \leftarrow L_\ell(x)$
- 19: **else if** L_ℓ is GT-Full (attention + FF + geometric transport) **then**
- 20: $A(z) \leftarrow \text{LN}_{\text{attn}}(z + \text{Drop}(\text{Attn}(z)))$
- 21: $F(z) \leftarrow \text{LN}_{\text{ff}}(z + \text{FF}(z))$
- 22: $G(z) \leftarrow \text{GeomTransport}(z)$ $\triangleright$ flatten $\rightarrow$ message pass $\rightarrow$ reshape
- 23: $\text{Obs}_\ell \leftarrow \frac{1}{3 \text{denom}} \left(\text{MSE}(A(F(x_{\text{in}})) - F(A(x_{\text{in}}))) + \right.$
 $\left. \text{MSE}(F(G(x_{\text{in}})) - G(F(x_{\text{in}}))) + \text{MSE}(A(G(x_{\text{in}})) - G(A(x_{\text{in}}))) \right)$
- 24: $x \leftarrow L_\ell(x)$
- 25: **else if** L_ℓ is GT-MoE (attention + MoE mixing) **then**
- 26: $A(z) \leftarrow \text{LN}_{\text{attn}}(z + \text{Drop}(\text{Attn}(z)))$
- 27: $M(z) \leftarrow \text{LN}_{\text{ff}}(z + \text{MoE}(z))$ $\triangleright$ gating + top- k + experts
- 28: $\text{Obs}_\ell \leftarrow \frac{\text{MSE}(A(M(x_{\text{in}})) - M(A(x_{\text{in}})))}{\text{denom}}$
- 29: $x \leftarrow L_\ell(x)$
- 30: **else**
- 31: $\text{Obs}_\ell \leftarrow \text{NaN}; \quad x \leftarrow L_\ell(x)$
- 32: **end if**
- 33: $\text{ObsList.append}(\text{Obs}_\ell)$
- 34: **end for**
- 35: $\text{Obs}(M; s) \leftarrow \text{mean}(\{\text{Obs}_\ell : \text{Obs}_\ell \text{ is finite}\})$
- 36: **Restore training mode; return** $\text{Obs}(M; s), \text{ObsList}$.

Algorithm 7: Diagrammatic Backpropagation Training Loop

Input: Model M_θ (Transformer or GT variant); dataset iterator $\mathcal{B}$ yielding minibatches (s, y) ; optimizer ADAMW; steps T ; warmup steps T_w ; weights $\lambda_{\text{db}}, \lambda_{\text{geom}}$; clipping norm c ; small constant ϵ

Output: Trained parameters θ

- 1: **Initialize:** set global step $t \leftarrow 0$
- 2: **for** $t = 1$ **to** T **do**
- 3: Sample minibatch $(s, y) \sim \mathcal{B} \triangleright s$: tokens (or inputs), y : labels/targets
- 4: **// 1) Forward pass for the primary task**
- 5: $\hat{y} \leftarrow M_\theta(s)$
- 6: $\mathcal{L}_{\text{task}} \leftarrow \text{LossTask}(\hat{y}, y) \quad \triangleright \text{e.g., cross-entropy for LM}$
- 7: **// 2) Optional DB control loss (commutator-energy terms)**
- 8: $\mathcal{L}_{\text{db}} \leftarrow 0$
- 9: **if** $\lambda_{\text{db}} > 0$ **then**
- 10: $\mathcal{L}_{\text{db}} \leftarrow \text{DIFFERENTIABLECOMMRESIDUAL}(M_\theta, s; \epsilon) \quad \triangleright$
 gradients enabled; not Algorithm 6
- 11: **end if**
- 12: **// 3) Optional registered-diagram loss (triangle/square residuals)**
- 13: $\mathcal{L}_{\text{geom}} \leftarrow 0$
- 14: **if** $\lambda_{\text{geom}} > 0$ **and** (s, y) includes a compiled diagram/simplicial instance **then**
- 15: $\triangleright$ For diagram benchmarks, s contains $(\mathcal{D}, x)$ with constraints.
- 16: $\mathcal{L}_{\text{geom}} \leftarrow \text{DIAGRAMRESIDUAL}(M_\theta, s) \quad \triangleright$ defined only for the
 declared paths and observations
- 17: **end if**
- 18: **// 4) Warmup schedule for auxiliary terms (optional but common)**
- 19: $\lambda_t^{\text{db}} \leftarrow \lambda_{\text{db}} \cdot \min(1, t/T_w)$
- 20: $\lambda_t^{\text{geom}} \leftarrow \lambda_{\text{geom}} \cdot \min(1, t/T_w)$
- 21: **// 5) Total loss and parameter update**
- 22: $\mathcal{L} \leftarrow \mathcal{L}_{\text{task}} + \lambda_t^{\text{db}} \mathcal{L}_{\text{db}} + \lambda_t^{\text{geom}} \mathcal{L}_{\text{geom}}$
- 23: Zero accumulated gradients; backpropagate $\mathcal{L}$
- 24: Clip gradients: $\|\nabla_\theta \mathcal{L}\| \leq c$
- 25: Update θ with one AdamW step
- 26: **end for**
- 27: **return** θ

Algorithm 8: GT-Full Geometric Transport Operator G (Simplicial/Graph Message Passing)

Input: Token states $z \in \mathbb{R}^{B \times L \times d}$; fixed adjacency $\text{edge\_index} \in \mathbb{N}^{2 \times E_{\max}}$ over positions $0, \dots, L_{\max} - 1$; relation ids $\text{rel\_ids} \in \mathbb{N}^{E_{\max}}$; domain ids $\text{dom\_ids} \in \mathbb{N}^{E_{\max}}$; depth K ; learnable parameters θ_G

Output: Transported token states $G(z) \in \mathbb{R}^{B \times L \times d}$

```

1: // 1) Restrict one positional graph to the current length
2:  $h \leftarrow \text{reshape}(z, (B \cdot L) \times d)$ 
3:  $m \leftarrow \{e : \text{edge\_index}[0, e] < L \wedge \text{edge\_index}[1, e] < L\}$   $\triangleright$ 
   restrict both endpoints
4:  $E_0 \leftarrow \text{edge\_index}[:, m]$ ;  $r_0 \leftarrow \text{rel\_ids}[m]$ ;  $d_0 \leftarrow$ 
    $\text{dom\_ids}[m]$ 
5:  $E \leftarrow \text{concat}_{b=0}^{B-1}(E_0 + bL)$ 
6:  $r \leftarrow \text{repeat}(r_0, B)$ ;  $dID \leftarrow \text{repeat}(d_0, B)$   $\triangleright B$  disjoint, index-offset
   graphs

7: // 2) Iterate  $K$  rounds of geometric message passing
8: for  $k = 1$  to  $K$  do
9:   Initialize message accumulator  $M \leftarrow 0 \in \mathbb{R}^{(BL) \times d}$ 
10:  for each directed edge  $(u \leftarrow v)$  in  $E$  do
11:     $e \leftarrow (u, v)$ 
12:    Edge embedding:  $e_{\text{emb}} \leftarrow \text{Emb}_{\text{rel}}(r_e) + \text{Emb}_{\text{dom}}(dID_e)$ 
13:    Compute message:  $m_{v \rightarrow u} \leftarrow \phi_{\theta}(h_v, h_u, e_{\text{emb}})$   $\triangleright$  e.g., MLP on
       $(h_v, h_u, e_{\text{emb}})$  or attention-style score
14:     $M_u \leftarrow M_u + m_{v \rightarrow u}$ 
15:  end for
16:  Degree normalize:  $M \leftarrow \text{DegNorm}(M, E)$ 
17:  Update rule:  $h \leftarrow \text{Update}_{\theta}(h, M)$   $\triangleright$  e.g., GRUCell/residual MLP +
    LayerNorm
18: end for

19: // 3) Reshape back to token grid
20:  $G(z) \leftarrow \text{reshape}(h, B \times L \times d)$ 
21: return  $G(z)$ 

```

Algorithm 9: GT-Lite Local Convolution Operator C (1D Conv + Residual + LN)

Input: Token states $z \in \mathbb{R}^{B \times L \times d}$, convolution weights θ_C ; smoothing scale α (e.g., 0.2); LayerNorm parameters θ_{LN}

Output: Smoothed token states $C(z) \in \mathbb{R}^{B \times L \times d}$

- 1: **// 1) Convert to Conv1d layout and apply local neighborhood mixing**
 - 2: $z^\top \leftarrow \text{transpose}(z, (B, L, d) \rightarrow (B, d, L))$
 - 3: $u^\top \leftarrow \text{Conv1D}_{\theta_C}(z^\top) \triangleright \text{kernel size 3, padding 1, output shape } (B, d, L)$
 - 4: $u \leftarrow \text{transpose}(u^\top, (B, d, L) \rightarrow (B, L, d))$
 - 5: **// 2) Residual update and normalization**
 - 6: $z' \leftarrow z + \alpha \cdot u$
 - 7: $C(z) \leftarrow \text{LayerNorm}_{\theta_{LN}}(z')$
 - 8: **return** $C(z)$
-

Algorithm 10: GT-MoE Mixing Operator M (Gating + Top- k + Expert Aggregation)

Input: Token states $z \in \mathbb{R}^{B \times L \times d}$; experts $\{E_e\}_{e=1}^E$; gating network $g_\theta : \mathbb{R}^d \rightarrow \mathbb{R}^E$; optional top- k value k ; LayerNorm parameters θ_{LN}

Output: Mixed token states $M(z) \in \mathbb{R}^{B \times L \times d}$

- 1: **// 1) Compute gating probabilities per token**
 - 2: $\ell \leftarrow g_\theta(z) \quad \triangleright \ell \in \mathbb{R}^{B \times L \times E} \text{ (gate logits)}$
 - 3: $p \leftarrow \text{softmax}(\ell, \text{ over experts}) \quad \triangleright p \in \mathbb{R}^{B \times L \times E}$
 - 4: **// 2) Optional sparsification (top- k routing) and renormalization**
 - 5: **if** $k < E$ **then**
 - 6: $(v, I) \leftarrow \text{TopK}(p, k) \quad \triangleright v, I \in \mathbb{R}^{B \times L \times k}$
 - 7: $\tilde{p} \leftarrow 0 \quad \triangleright \text{initialize } \tilde{p} \in \mathbb{R}^{B \times L \times E}$
 - 8: $\tilde{p}[\cdot, \cdot, I] \leftarrow v \quad \triangleright \text{scatter top-}k \text{ weights into expert slots}$
 - 9: $p \leftarrow \tilde{p} / (\sum_{e=1}^E \tilde{p}_e + \epsilon) \quad \triangleright \text{renormalize per token}$
 - 10: **end if**
 - 11: **// 3) Apply experts and aggregate weighted outputs**
 - 12: $u \leftarrow 0 \in \mathbb{R}^{B \times L \times d}$
 - 13: **for** $e = 1$ **to** E **do**
 - 14: $u \leftarrow u + p_e \odot E_e(z) \quad \triangleright p_e \text{ broadcast to shape } (B, L, d); \odot \text{ elementwise}$
 - 15: **end for**
 - 16: **// 4) Residual update and normalization**
 - 17: $z' \leftarrow z + u$
 - 18: $M(z) \leftarrow \text{LayerNorm}_{\theta_{LN}}(z')$
 - 19: **return** $M(z)$
-

Interpreting the Experiments

The baseline, GT-Lite, GT-Full, and GT-MoE models exhibit different trajectories. The statistic directly measures forward order sensitivity for the registered operators on the sampled states. It does not identify the architectural mechanism responsible for differences in that statistic, which remains an empirical question.

Reframing the obstruction proxy

Operationally, the plotted quantity was an average of normalized commutator energies:

$$\text{Comm}(T_1, T_2; x) = \frac{\|T_1(T_2(x)) - T_2(T_1(x))\|_2^2}{\|x\|_2^2 + \varepsilon}.$$

High energy means that the two counterfactual forward compositions differ on the chosen states; low energy means that they are approximately exchangeable there. Neither statement entails a Čech cocycle, parameter-update exchangeability, improved optimization, or improved generalization.

Baseline growth

The baseline run exhibited increasing forward order sensitivity alongside worse validation performance. This is a correlation from one experimental protocol, not an intrinsic theorem about Transformers. Depth, parameter count, normalization, optimization, and information flow are potential confounders. Testing whether the statistic predicts instability requires matched seeds and architectures, direct gradient-conflict measurements, and interventions that change the residual while holding the information regime fixed.

GT-Lite: partial mitigation via local smoothing

GT-Lite reduced the statistic early and later drifted. Local convolutional mixing is one candidate explanation, but a learned full-channel kernel need not be smoothing or low-pass. Residual scale, normalization, and—in decoder experiments—future leakage from symmetric padding are alternative explanations. A causal comparison must register these choices explicitly.

GT-Full: transport as a candidate mechanism

GT-Full maintained lower, more stable values in the reported runs. Graph transport is a plausible mechanism, but low residuals are not structurally inevitable and do not certify coordinate alignment or reduced curvature. The displayed implementation also requires a corrected batched graph and a separate causal information-flow analysis. Matched transport ablations and collapse controls are the appropriate admission tests.

GT-MoE: conditional computation and intermediate regimes

GT-MoE produced intermediate values. The routing hypothesis predicts that specialization and top- k boundary effects pull the statistic in different directions. Routing-flip measurements and gate ablations can test that prediction; the observed ordering alone cannot.

SUMMARY. The experiments establish a useful diagnostic pattern, not a completed causal explanation. Their enduring contribution is to make operator order sensitivity measurable and to identify architecture-specific hypotheses that can be tested with matched controls.

Frontier Case Study: Kimi K3 as a Compositional Architecture

Kimi K3 provides a snapshot, as described in its cited technical report, of why compositional architecture has become a practical concern at the frontier of machine learning. The open-weight model has 2.8 trillion total parameters but activates approximately 104 billion parameters for each token. Its technical report does not describe scale as one homogeneous operation. Instead, it organizes the architecture around information flow across sequence length, network depth, and model width [Kimi Team, 2026].

Along the sequence axis, each repeated block combines three recurrent Kimi Delta Attention layers with one Gated MLA layer. The recurrent layers provide efficient position-sensitive mixing; the periodic MLA layer restores unrestricted global content interaction. Along the depth axis, Attention Residuals allow a layer to retrieve selected representations from earlier blocks rather than receiving only a uniformly accumulated residual state. Along the width axis, Stable LatentMoE retains a full-width shared route while dispatching a lower-dimensional latent representation to a sparse selection of specialized experts. These are three different compositions with three different interfaces.

The architecture can therefore be read as a typed diagram rather than merely as a large parameter tensor. If K denotes a KDA layer and G a Gated MLA layer, a repeated sequence-mixing block has the ordered form

$$S = G \circ K_3 \circ K_2 \circ K_1.$$

If $P_\downarrow$ and $P_\uparrow$ move between the full model width and the routed latent width, then the routed branch has the schematic form

$$M(x) = P_\uparrow \left(\sum_{i \in \text{TopK}(x)} g_i(x) E_i(P_\downarrow x) \right),$$

combined with a full-width shared-expert path. The router $g_i(x)$ makes the realized morphism input dependent, exactly the source of order sensitivity identified in the GT-MoE analysis. Attention Residuals add a third family of

paths from earlier block representations to the current layer. A change in one family can alter the effective inputs seen by the others.

The following comparison separates the interfaces and the compatibility question attached to each one.

Architectural family	Categorical reading	Compatibility question
KDA and Gated MLA	Ordered composition of recurrent and global sequence operators	Does efficient local state retain what periodic global retrieval requires?
Attention Residuals	A learned family of paths from earlier blocks to a later representation	Which depth-wise paths may be summarized without losing a needed representation?
Stable LatentMoE	An indexed expert family selected by a state-dependent router	Does attention change the active expert family, and does routing change the representation attended to?
Low-precision realization	A comparison between high- and low-precision implementations of registered subdiagrams	Does quantized execution preserve the observations used to judge the model?
Distributed execution	A compilation of one abstract graph into pipeline, expert, data, and context-parallel operations	Do alternative partitions realize the same declared computation within tolerance?

This is a categorical interpretation of the published architecture, not a claim that Moonshot designed the model using category theory.

Quantization-aware post-training supplies a particularly clear preservation question. Kimi K3 quantizes the dominant routed-expert weights to MXFP4 and computes their activations in MXFP8, while keeping routers, shared experts, attention projections, and latent projections at higher precision. Training and rollout use the same registered quantization scheme during reinforcement learning [Kimi Team, 2026]. The intended comparison is not that the two numerical realizations are literally equal. It is that the high-precision and deployment paths agree under the task observers and tolerances declared for the system. Quantization-aware training attempts to repair the failure of this approximate square to commute.

This example also places the smaller GT-MoE experiments in perspective. The toy commutator

$$A(M(x)) - M(A(x))$$

does not explain Kimi K3's capabilities, nor does it certify the complete architecture. It isolates one interaction that becomes more consequential, not less, when a system contains hundreds of experts, multiple attention mechanisms, depth-wise retrieval, mixed numerical precision, and distributed execution. Category theory contributes a language for keeping those interfaces and preservation questions distinct.

Design principle

A frontier model should be read at two levels: as a learned function and as a diagram of interacting realizations. Parameter count describes the size of the first representation; it does not certify the interfaces of the second.

Conclusion: What the Probe Does and Does Not Establish

The commutator energy in this chapter is a forward, distribution-relative probe of order sensitivity. It can compare registered pairs of operators and can be used as a differentiable auxiliary objective when recomputed with gradients enabled. It does not automatically measure a Čech obstruction, curvature, gradient interference, optimization stability, expressivity, or generalization. Each stronger interpretation requires its own declaration and admission test.

The code audit also exposed why those distinctions matter. The displayed GT-Lite decoder convolution uses symmetric padding and therefore violates strict autoregressive information flow. The displayed GT-Full transport flattens a batch without replicating and offsetting its graph. The GT-Lite convolution is full-channel rather than depthwise in this implementation. These are not minor presentational details: they change the realized diagram.

The empirical ordering among the baseline, GT-Lite, GT-MoE, and GT-Full models therefore motivates hypotheses rather than proving mechanisms. Local convolution, graph transport, and expert routing are three separable architectural interventions. Their effects should be evaluated with matched capacity, information regime, residual scale, normalization, and seeds, together with direct checks for collapse and gradient conflict.

The next chapter makes one of these controls explicit by separating architecture class from information regime. That distinction is indispensable whenever an auxiliary geometric path can carry information unavailable to the declared baseline.

Suggested Reading

For the differential and topological background behind commutators, curvature, and obstruction, see [Bott and Tu \[1982\]](#), [Ghrist \[2014\]](#), and [Edelsbrunner et al. \[2002\]](#). The compositional-learning viewpoint begins with functorial backpropagation [[Fong et al., 2019](#)]. For a current primary account of a frontier architecture organized across sequence, depth, width, numerical precision, and distributed execution, see the Kimi K3 technical report [[Kimi Team, 2026](#)]. [Dickson \[2026\]](#) gives a shorter production-oriented overview; technical claims should be checked against the primary report.

Information Regimes in Geometric Transformers

This chapter separates two questions previously conflated in the GT studies: *architecture class* and *information regime*. Causal self-attention does not make the entire model causal if a convolutional, geometric, or predictive side branch reads a future-indexed state. The audit therefore follows computational provenance all the way to each target logit. It also treats gradient flow as a separate axis: `detach` changes which parameters receive derivatives, but it does not change what information the forward computation consumes.

Information Regimes as Typed Provenance

For target position t , let $\mathcal{C}_{\leq t}$ denote the typed subcategory of the realized computation graph generated by inputs $x_0, \dots, x_t$, model parameters, and operations registered as available at inference time. A target logit is strictly causal when every path into it lies in this prefix-generated subcategory. This is a provenance condition, not the claim that an ordinary Transformer is literally a functor from the thin poset of token indices.

We distinguish three labels:

- **C**: prefix-only computation with no predictive side carrier;
- **E**: endogenous self-conditioning whose carrier is itself computed entirely inside $\mathcal{C}_{\leq t}$;
- **A**: augmented computation with at least one path from a future-indexed or target-revealing object.

Regimes C and E obey the same autoregressive information budget, but remain separate because self-conditioning changes the realized architecture. We analyze six implementation variants:

- `transformer_causal`
- `transformer_future_hint`
- `gt_causal`
- `gt_noncausal`
- `gt_pred_next_detach`

- `gt_pred_prev_causal_detach`

and report each one with an explicit regime label and side-channel description.

FUTUREHINT VS PREDICTIVEHINT. `transformer_future_hint` injects an explicit symmetric future-informed embedding path. The reported `gt_pred_next_detach` run first constructs predicted embeddings and then applies a *noncausal* kernel-3 convolution. Consequently, its update at t reads the predicted carrier at $t + 1$, which was computed from a hidden state containing teacher-forced x_{t+1} . It is therefore Regime A despite the detach. A hypothetical same-index predictive carrier followed only by left-padded mixing would instead be Regime E.

A Kan-Style Analogy—and Its Limit

Kan extensions are developed precisely in the next chapter. They motivate an analogy here, but do not follow merely from adjoining the next token position. Indeed, if $\mathcal{I}$ has arrows $i \rightarrow j$ for $i \leq j$ and $\iota_t : \mathcal{I}_{\leq t} \hookrightarrow \mathcal{I}_{\leq t+1}$, then the comma category $(t + 1 \downarrow \iota_t)$ is empty. In **Vect** the right Kan extension at $t + 1$ is therefore the terminal zero object, not a learned next-token prediction. A nontrivial Kan construction requires a richer observation category, reversed variance, or additional extension data.

The operational self-conditioning pattern studied here is simply:

1. predict a next-token distribution from prefix-derived hidden states;
2. map this prediction into an embedding-level auxiliary branch;
3. transport those carriers through a declared mixer.

Whether the result lies in E or A depends on the indices subsequently mixed, not on the word “predicted” and not on detachment. Calling such a mechanism Kan-style is a design interpretation until an indexing category, functors, and universal property are supplied.

The Geometric Transformer

GT augments Transformer-style mixing with a local branch over sequence positions. Calling positions vertices and adjacent pairs edges supplies a graph index; the implementation in this chapter is a depthwise Conv1D rather than a higher-dimensional simplicial operator. The architectural plate below separates the baseline, left-padded causal convolution, symmetric noncausal convolution, and predictive-carrier variants.

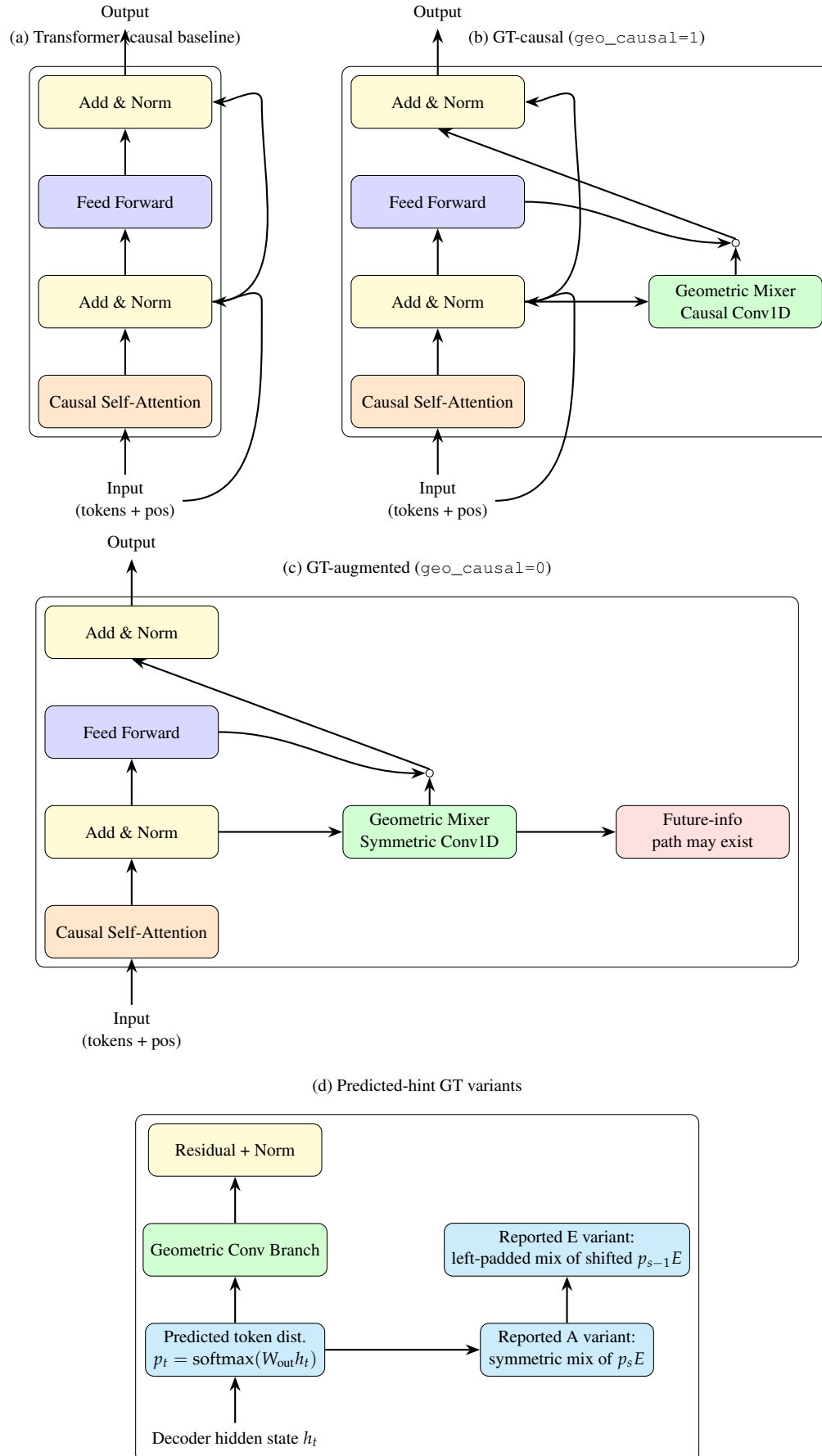


Figure 21: GT architectural variants studied in this chapter.

Categorical Interpretation of Auxiliary Channels

The central issue is not whether attention is causal, but whether the *full update diagram* at index t includes extra morphisms that are future-informative. We formalize this at the diagram level.

NOTATION. For a token sequence, let X_i denote the input representation at position i , let H_t be the hidden state used to produce target logit $\hat{y}_t$, and let P_t be the subgraph of all registered computations whose source indices are at most t . A Transformer does not literally compute through the linear chain $X_0 \rightarrow \cdots \rightarrow X_t$; masked attention forms a richer dependency graph. What matters is that the backward provenance cone of $\hat{y}_t$ is contained in P_t .

BASE CAUSAL DIAGRAM. Regard the realized tensor program as a typed directed acyclic graph. The strict-causal declaration requires a factorization

$$(X_0, \dots, X_t) \longrightarrow H_t \longrightarrow \hat{y}_t$$

inside P_t . No colimit or universal property is implied by this dependency factorization.

AUXILIARY AUGMENTATION. Suppose we add an auxiliary object A_t and morphism

$$\alpha_t : A_t \rightarrow H_t,$$

where H_t is the decoder state at position t . This extends the diagram from a prefix-only shape to a larger category $\tilde{\mathcal{L}}_t$.

Definition 42 (Admissible Causal Augmentation). *An auxiliary map α_t is admissible causal iff its realized provenance lies in P_t . Equivalently, for the registered tensor program there is a prefix-derived object G_t and maps*

$$A_t \rightarrow G_t \rightarrow H_t$$

with G_t constructed only from inputs and operations available at indices $\leq t$.

Definition 43 (Future-Informative Augmentation). *An auxiliary map α_t is future-informative when its realized provenance includes an input indexed by $> t$ or another carrier that reveals the target under the evaluation protocol.*

These are implementation-relative definitions. An abstract map may admit many factorizations; causal status is determined by the realized computation and its train/test alignment, not by the mere existence of some set-theoretic factorization.

Operationally, this means causal self-attention by itself is insufficient to guarantee strict causal evaluation: the geometric branch can alter the information regime depending on the source of its inputs.

$$X_0 \rightarrow X_1 \rightarrow \cdots \rightarrow X_t \rightarrow H_t \rightarrow \hat{y}_t$$

(a) Base causal diagram (prefix-only).

$$\begin{array}{c} X_0 \rightarrow \cdots \rightarrow X_t \rightarrow G_t \xrightarrow{\gamma_t} H_t \rightarrow \hat{y}_t \\ \beta_t \uparrow \quad \quad \quad \alpha_t \nearrow \\ A_t \end{array}$$

(b) Admissible causal augmentation: $\alpha_t = \gamma_t \circ \beta_t$.

$$\begin{array}{c} X_0 \rightarrow \cdots \rightarrow X_t \rightarrow H_t \rightarrow \hat{y}_t \\ X_{t+1} \rightarrow A_t \quad \quad \quad \alpha_t \nearrow \end{array}$$

(c) Future-informative augmentation: A_t depends on X_{t+1} .

Figure 22: **Information-regime diagrams.** Prefix-only computation (left); a prefix-derived auxiliary carrier, admissible as C or E (middle); and a future-informed carrier in A (right).

Model-to-regime mapping by auxiliary morphism

Variant	Auxiliary map type	Regime
transformer_causal	none beyond prefix functor	C
gt_causal	causal local map (left-padded)	C
gt_pred_prev_causal_detach	shifted, left-padded self-conditioning	E
gt_noncausal	symmetric local map (can mix $t \pm 1$)	A
transformer_future_hint	explicit future-informative hint path	A
gt_pred_next_detach	predicted carriers + symmetric convolution	A

Language-Model Instantiation

For token sequence length L :

- token positions $1, \dots, L$ are 0-simplices;
- adjacency $(i, i + 1)$ defines 1-simplices;
- global operator is causal self-attention;
- geometric operator is a local 1D mixer (causal or symmetric depending on regime).

The implementation first chooses a carrier at every source position s :

$$\tilde{u}_s \in \{h_s, \text{detach}(p_s E), \text{detach}(p_{s-1} E)\},$$

$$p_s = \text{softmax}(W_{\text{out}} h_s / \tau),$$

and then applies either a left-padded or symmetric depthwise convolution across s . Thus the target update is causal only if every carrier inside its kernel window has prefix provenance relative to that target. The three alternatives are hidden, same-index predicted, and shifted predicted carriers.

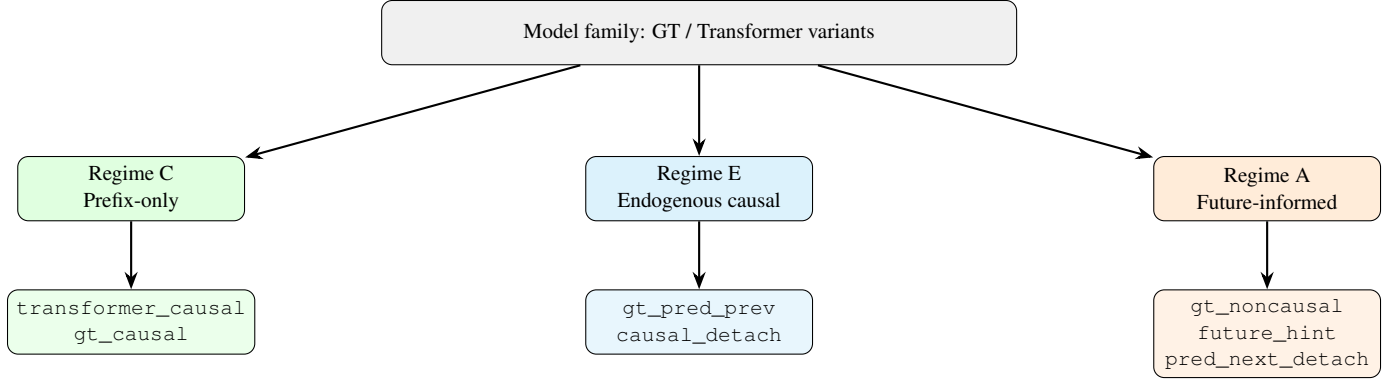
WHICH GT FAMILY IS ABLATED HERE? All variants in this leakage protocol are GT-Lite style instantiations (Transformer + local depthwise Conv1D geometric mixer). They do not include GT-Full simplicial transport blocks or GT-MoE routing. This is intentional: the protocol isolates information-regime effects without adding capacity/routing confounds from the larger GT families.

*Revised LM Interpretation***Experiment: GT information-regime ablation**

The implementation audit yields the following classification:

- Regime C contains `transformer_causal` and `gt_causal`.
- Regime E contains `gt_pred_prev_causal_detach`; it remains prefix-only but adds an endogenous predictive carrier.
- Regime A contains `gt_noncausal`, `transformer_future_hint`, and the reported `gt_pred_next_detach`, all of which possess a forward path from a future-indexed teacher-forced state.

Near-unit Regime-A perplexity is therefore a leakage diagnostic, not a language modeling gain.



REPORTING STANDARD. Each LM table should include the C/E/A label, `geo_causal`, carrier source, shift policy, target alignment, and whether detachment is enabled.

WikiText-103 and WikiText-2 Results by Regime

This section reports recorded endpoint snapshots for PTB, WikiText-2, and WikiText-103, grouped by realized provenance rather than model name alone. They are point estimates from single runs; no multi-seed uncertainty is available in these tables. The interpretation standard is:

1. compare autoregressive performance only among C and E variants whose target-relative provenance has been verified;
2. treat A results as positive leakage controls, not competing language models;
3. require matched seeds and uncertainty before treating small within-regime differences as robust architecture effects.

WIKITEXT-2 (L=2, D=256) SNAPSHOT. The recorded WikiText-2 run shows C and E variants near one another, while A variants obtain much lower teacher-forced perplexity by accessing future-informed paths. In this table, all `gt_*` entries are GT-Lite style models (Transformer + local depthwise Conv1D geometric branch), not GT-Full or GT-MoE.

Model	Regime	Val PPL	Test PPL
<code>transformer_causal</code>	C	178.857	163.530
<code>gt_causal</code>	C	171.704	160.193
<code>gt_pred_prev_causal_detach</code>	E	174.516	160.406
<code>transformer_future_hint</code>	A	2.053	2.010
<code>gt_pred_next_detach</code>	A	1.737	1.696
<code>gt_noncausal</code>	A	1.051	1.051

Figure 23: **Evaluation taxonomy by information budget.** C and E share a causal information budget but differ architecturally. Regime A uses target- or future-informed paths and its perplexities are not comparable to autoregressive perplexity.

Table 4: WikiText-2 provenance-ablation results grouped by information regime. Values for A variants are teacher-forced leakage diagnostics, not autoregressive perplexities.

PTB ($L=2$, $D=256$) SNAPSHOT. PTB shows the same qualitative diagnostic: C and E variants cluster, whereas A variants approach trivial teacher-forced reconstruction.

Model	Regime	Val PPL	Test PPL
transformer_causal	C	134.030	124.793
gt_causal	C	137.483	128.364
gt_pred_prev_causal_detach	E	134.129	124.487
transformer_future_hint	A	1.521	1.384
gt_pred_next_detach	A	1.099	1.054
gt_noncausal	A	1.075	1.050

Table 5: PTB provenance-ablation results grouped by information regime. Values for A variants are teacher-forced leakage diagnostics, not autoregressive perplexities.

THE DELAYED-LEAKAGE VARIANT ($GT\_PRED\_NEXT\_DETACH$).

This variant often begins near the causal models and later moves toward the future-informed controls. The implementation explains why. Early predicted carriers are poor, so the symmetric convolution has little useful information to transport from $t + 1$ into t . As those carriers improve, the path from the teacher-forced state at $t + 1$ becomes increasingly informative. The curve is therefore evidence that the model learns to exploit a leaky path; it is not evidence for causal self-conditioning. The E control is `gt_pred_prev_causal_detach`.

SMALLER-MODEL CHECK ($L=2$, $D=64$). At lower width, `gt_pred_next_detach` exploits the future-informed path less completely. On PTB its test perplexity is 16.567, compared with ~ 208 – 222 for C/E models and ~ 1.144 – 4.940 for the other A controls. On WikiText-2 it reaches 44.050, between ~ 288 – 306 for C/E and ~ 1.421 – 10.160 for the other A controls. This is compatible with capacity-dependent leakage exploitation, though multiple seeds are needed.

Dataset	Model	Regime	Test PPL
PTB ($d = 64$)	transformer_causal	C	222.238
PTB ($d = 64$)	gt_causal	C	207.721
PTB ($d = 64$)	gt_pred_prev_causal_detach	E	216.909
PTB ($d = 64$)	gt_pred_next_detach	A	16.567
PTB ($d = 64$)	transformer_future_hint	A	4.940
PTB ($d = 64$)	gt_noncausal	A	1.144
WikiText-2 ($d = 64$)	transformer_causal	C	305.511
WikiText-2 ($d = 64$)	gt_causal	C	288.030
WikiText-2 ($d = 64$)	gt_pred_prev_causal_detach	E	295.857
WikiText-2 ($d = 64$)	gt_pred_next_detach	A	44.050
WikiText-2 ($d = 64$)	transformer_future_hint	A	10.160
WikiText-2 ($d = 64$)	gt_noncausal	A	1.421

Table 6: Small-model ($L = 2$, $d = 64$) provenance-ablation snapshot. Values for A variants are teacher-forced leakage diagnostics.

LEAKAGE-EXPLOITATION INDEX. Define

$$\text{leakage\_proximity} = \frac{\text{PPL}_{\text{C/E best}} - \text{PPL}_{\text{pred\_next}}}{\text{PPL}_{\text{C/E best}} - \text{PPL}_{\text{A best}}},$$

where the denominator uses the best recorded future-informed control. This is a descriptive index of how completely one A variant exploits its leakage path. It is not a measure of causal self-conditioning or model quality, and it need not lie in $[0, 1]$ outside the recorded ordering.

Dataset	d	C/E best	Pred-next	Leakage proximity
PTB	256	124.487	1.054	1.000
WikiText-2	256	160.193	1.696	0.996
PTB	64	207.721	16.567	0.925
WikiText-2	64	288.030	44.050	0.851

WIKITEXT-103 ($L=2$, $D=256$) SNAPSHOT. The recorded WikiText-103 run shows the same provenance split: C and E variants cluster together, while A variants obtain much smaller teacher-forced losses. The delayed-leakage variant exploits its path less completely than the direct future controls at this endpoint.

Model	Regime	Val PPL	Test PPL
transformer_causal	C	234.666	232.674
gt_causal	C	220.219	215.384
gt_pred_prev_causal_detach	E	228.757	223.917
transformer_future_hint	A	2.728	2.825
gt_pred_next_detach	A	8.748	8.748
gt_noncausal	A	1.215	1.267

Table 7: Descriptive proximity of one future-informed variant to the strongest future-informed control. The index quantifies leakage exploitation, not valid autoregressive performance.

Table 8: WikiText-103 provenance-ablation results grouped by information regime. Values for A variants are teacher-forced leakage diagnostics, not autoregressive perplexities.

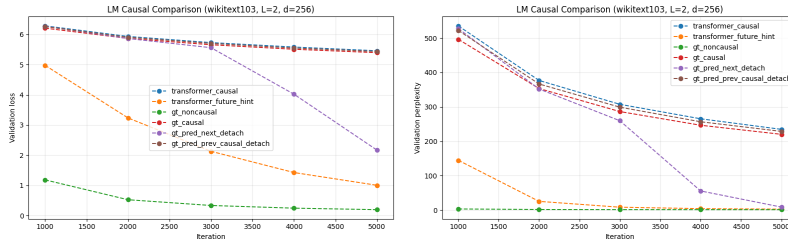


Figure 24: WikiText-103 provenance-ablation curves ($L = 2$, $d = 256$). The delayed exploitation of the future-informed path in `gt_pred_next_detach` is clearer in the trajectory than in the endpoint table. A-variant curves are leakage controls, not autoregressive performance results.

Interpretation and Scope

The principal result is methodological. Architecture labels do not determine causal validity; the full provenance graph does. Within the recorded C/E runs, differences are modest and lack multi-seed uncertainty. The dramatic A results show that several architectures can exploit future information and should not be compared with autoregressive perplexity.

The next chapter develops Kan extensions and coend-style aggregation. The evidence supplies a non-negotiable constraint for those constructions: universal or geometric language does not excuse an implementation whose target update depends on a future-indexed carrier.

Suggested Reading

The architectural baseline is the Transformer of [Vaswani et al. \[2017\]](#). The role of normalization and information pathways is illuminated by [Xiong et al. \[2020\]](#) and [Kim et al. \[2025\]](#). Retrieval-augmented context provides a useful comparison [[Khandelwal et al., 2020](#)].

Kan Extension and Topological Coend Transformers

This chapter develops two categorical generalizations of Transformer architectures: *Kan Extension Transformers* (KET) and *Topological Coend Transformers* (TopoCoend). Both arise from the same underlying problem: *how should local token representations be extended to larger structured domains in a principled way?* At the level of sets, this question is fundamentally underdetermined. A function specified on a finite sample admits many non-equivalent extensions to a larger domain, with no canonical choice in general. By contrast, once representations are organized functorially and related by an explicit structural map, category theory provides universal extension operators—left and right Kan extensions—that make precise how information should be aggregated or propagated. This shift from ad hoc mixing rules to candidate universal extension principles is the conceptual starting point of the chapter. A neural weighted sum is not a Kan extension merely because it resembles a pointwise formula; the indexing categories, functors, natural transformation, and universal property must still be supplied.

Our first construction, *Kan Extension Transformers*, interprets Transformer-style contextualization as a structured extension problem. Attention, convolutional mixing, and higher-order message passing can all be viewed as aggregation over neighborhoods of tokens, edges, or motifs. KET explores this idea by organizing token representations over a finite simplicial indexing structure whose objects encode vertices, edges, and higher-dimensional incidence, and by treating contextual updates as Kan-style weighted aggregations along simplicial incidence maps. Such an aggregation is an actual enriched left Kan extension only when the weights arise from an appropriate enriched hom-object (or representable profunctor) and satisfy the required functoriality and naturality conditions. Under this view, a Transformer layer is no longer merely a sequence model: it becomes a functorial mechanism for extending local token information across a structured combinatorial domain.

Our second construction, *Topological Coend Transformers*, provides a complementary realization of the same general idea. Instead of fixing the neighborhood structure combinatorially in advance, TopoCoend learns a low-dimensional coordinate embedding

$$z_t = \pi(h_t)$$

or, in self-conditioned regimes, $z_t = \pi(\hat{e}_t)$, and uses this embedding to define a weighted k -nearest-neighbor graph. Aggregation is then performed by a coend-style weighted sum over these learned neighborhoods, yielding a geometric message-passing update inside each Transformer block. The difference between KET and TopoCoend is therefore not the abstract form of aggregation itself, but the categorical origin of the neighborhoods over which the aggregation is taken: simplicial incidence in KET, learned latent-space adjacency in TopoCoend.

A central theme of the chapter is that *information regime* matters at least as much as architectural form. Naive noncausal geometric aggregation, when performed under teacher forcing, allows hidden states to encode gold future tokens and leads to pathological perplexity collapse. By contrast, replacing gold future values with *detached predictive distributions* blocks gradients through the auxiliary predictive branch and avoids directly transporting teacher-forced hidden states. Detachment alone, however, does not establish strict autoregressive validity. For an update at target t , every carrier it consumes must be measurable with respect to the target prefix $x_{\leq t}$. A carrier computed at a future source $s > t$ from $x_{\leq s}$ fails this target-relative filtration condition even when it is detached. We therefore describe such experimental variants as *noncausal self-conditioning*, reserving *strict-causal* for variants satisfying the target-relative condition.

We study both quadratic and incidence-restricted KET variants, together with intended-causal and future-carrier TopoCoend models, on Penn Treebank, WikiText-2, and WikiText-103. The recorded strict-causal point estimates show KET models competitive with Geometric Transformers (GT) and standard Transformers; quadratic KET has the lowest recorded strict-causal perplexity on WikiText-2 and WikiText-103, while incidence-restricted KET is less computationally expensive. These are single-run comparisons, not estimates of a population effect. The much larger numerical changes in the reported future-carrier variants occur in Regime A and are retained as leakage diagnostics rather than language modeling improvements. This sharp separation between *gold noncausal mixing*, *strict-causal modeling*, and *noncausal predictive self-conditioning* is one of the main lessons of the chapter.

More broadly, the constructions suggest a conceptual comparison. Attention, geometric mixing, Kan-style aggregation, and coend-style aggregation should not be viewed as unrelated architectural tricks. They share a weighted-aggregation template that can be compared with universal constructions for extending structured representations. The theoretical program suggested by this chapter asks whether a restricted class of local structured aggregation operators can be characterized by weighted Kan extensions, and how learned metric neighborhoods might fit such a characterization. No universality theorem is claimed for the generic neural implementations. Universal constructions instead provide a precise comparison target and a language for separating verified structure from architectural analogy.

Background

Autoregressive language models predict tokens one step at a time from a prefix. This strict causal discipline is both their strength and their limitation. It yields a valid factorization of the joint distribution, but it also restricts the model to a one-pass left-to-right flow of information. As we saw in the previous chapter, Geometric Transformer (GT) models suggest that *self-conditioning* can partially overcome this restriction. When a model first generates candidate continuations and then conditions on them in a subsequent pass, outputs effectively become new inputs. A simple and now empirically visible example is *prompt repetition*, where repeating a prompt or partial completion improves downstream responses because the model is allowed to condition on additional derived context ¹⁷.

GT models enrich the ordinary semantic branch of an autoregressive Transformer with an additional *geometric mixing* branch that propagates information along an explicit incidence structure, such as the 1-skeleton of a token sequence. This added branch changes the model’s inductive bias and may alter empirical performance, but it also raises a critical question: *when does nonlocal or noncausal mixing become equivalent to future-token leakage?* Under teacher forcing, hidden states may encode gold future tokens. If a geometric branch accesses such states noncausally, even weak long-range couplings can cause dramatic and misleading perplexity collapse. The resulting models may look extraordinarily good numerically while being invalid as autoregressive models.

This chapter develops a categorical lens that clarifies these issues. The key shift is from thinking of a layer as a hand-designed mixing rule to thinking of it as a *structured extension operator*. Given local token representations on a smaller domain, how should one extend them to richer domains built from edges, motifs, simplices, or learned neighborhoods? On sets alone, there is no canonical answer. But once the problem is formulated functorially, universal constructions enter naturally: left Kan extensions describe universal aggregation from local evidence once a functorial extension problem has been specified, while coends express the corresponding identifications. Generic neighborhood-wise weighted sums are only analogues until those data and laws are verified. In this sense, the chapter studies not just new architectures, but a new design language for Transformer-style models.

CHAPTER PERSPECTIVE. The present chapter focuses primarily on the *left-Kan* side of the story: contextualization, aggregation, and geometric mixing. Later developments connect these ideas to complementary notions of structured completion and refinement. For language modeling, however, the immediate question is already rich: how should a model aggregate over structured neighborhoods without violating causal validity?

¹⁷ Yaniv Leviathan, Matan Kalman, and Yossi Matias. Prompt repetition improves non-reasoning llms, 2025. URL <https://arxiv.org/abs/2512.14982>

Autoregressive Transformers

Let $x_{0:S-1}$ denote a token sequence. A causal Transformer computes hidden states

$$h_t = f_\theta(x_{\leq t})$$

using masked self-attention, feedforward sublayers, and residual updates. The model defines next-token distributions

$$p_\theta(x_{t+1} \mid x_{\leq t}) = \text{softmax}(\ell_t),$$

where $\ell_t = W_o h_t$ are the logits produced from the final hidden state.

Strict causality requires that h_t depend only on the prefix $x_{\leq t}$. During teacher-forced training, the hidden state may encode gold prefix information, but it must not encode gold future tokens if the resulting model is to define a valid autoregressive factorization. This distinction becomes crucial once additional nonlocal mixing mechanisms are introduced.

Geometric Mixing in Transformers

Geometric Transformers augment the semantic branch of a Transformer with an additional geometric operator $g(h)$ defined from an incidence structure on the sequence. A schematic GT layer takes the form

$$h \mapsto \text{LN}(h + \text{Attn}(h) + g(h)).$$

Here $g(h)$ may represent local convolution, message passing over an adjacency graph, or a more general incidence-based update. The important point is that information is no longer propagated only through masked self-attention. It is also propagated through an explicitly specified geometric neighborhood system. This introduces a relational inductive bias and, empirically, can improve both stability and perplexity.

Why Kan Extensions Enter

The transition from GT to KET begins with a simple conceptual question: what is the right way to extend local token information across a structured neighborhood system?

For ordinary functions on sets, this extension problem is typically non-canonical. Samples on a smaller domain admit many incompatible extensions to a larger one: constant predictors, interpolants, splines, kernels, and so on. What category theory adds is not a magical elimination of modeling choices, but a disciplined relocation of those choices into the *structure* of the extension problem itself. Once local values are organized as a functor

$$F : A \rightarrow \mathcal{E}$$

and related to a larger target structure by a functor

$$K : A \rightarrow B,$$

universal constructions become available. A left Kan extension $\text{Lan}_K F$ gives a principled way to aggregate local contributions into values on B . In enriched settings, this aggregation becomes weighted, making it a natural language for attention-style contextualization.

This perspective is especially useful for Transformer models. Tokens may be treated as 0-simplices, adjacent pairs as 1-simplices, and larger motifs as higher-dimensional simplices. A contextual update at a token can then be viewed as an extension from values defined on simplices back to values on tokens. Under this reading, attention and geometric mixing are no longer fundamentally different mechanisms: they are both structured extension operators, differing only in the choice of neighborhood system and weights.

Information Regimes

In earlier work on Geometric Transformers, we identified several distinct *information regimes* governing the interaction between geometric mixing and autoregressive training:

- **Strict causal:** all geometric updates at position t depend only on states derived from $x_{\leq t}$.
- **Gold noncausal (invalid):** geometric updates use hidden states that encode gold tokens at positions $> t$, causing future-token leakage.
- **Predict-and-detach (self-conditioned):** geometric updates use detached predictive values. This label describes carrier construction, not causal validity; target-relative provenance must be checked separately.

In the predict-and-detach regime, the model replaces unavailable future gold information with its own predictive distributions:

$$\hat{e}_t = \text{detach}\left(\text{softmax}(\ell_t/T) E\right),$$

where E is the embedding matrix. Because $\hat{e}_t$ is derived from prefix-conditioned logits, it may be used at target u only when its provenance lies in $x_{\leq u}$. Detachment does not relax that condition. In particular, a carrier at $t > u$ computed from a teacher-forced state exposes information unavailable at target u .

Empirically, the distinction is stark. Gold noncausal models collapse to perplexity near 1, revealing invalid future leakage. Strict-causal models remain valid but may underuse structure. The reported future-carrier predict-and-detach models also obtain very low teacher-forced perplexity, but the implementation audit places them in Regime A; those values are therefore leakage diagnostics rather than evidence of causal legitimacy. This separation between information regimes is central to every architectural claim made in the chapter.

From GT to KET and TopoCoend

KET enlarges the GT design space by replacing one hand-specified geometric mixing pattern with an explicit Kan-style aggregation operator. The basic ingredients are: tokens as 0-simplices, incidence relations defining higher-order structure, and a weighted extension from simplices back to token states. Two realizations are studied in this chapter.

The first is a *quadratic* KET variant, which performs global kernel-style pooling over simplices and is therefore analogous to a second attention-like aggregation branch. The second is an *incidence-restricted* KET variant, which aggregates only over incident simplices and therefore reduces to a linear-time message-passing scheme. This incidence-restricted form makes the computational relation to GT especially transparent: both use local incidence, although an exact special-case theorem would require a parameter map identifying their operators.

TopoCoend provides a different path to structured aggregation. Instead of fixing simplicial neighborhoods combinatorially, it learns a low-dimensional coordinate map $z_t = \pi(h_t)$ and uses this map to construct weighted k -nearest-neighbor neighborhoods. Aggregation is then computed by a coend-style weighted sum over those neighborhoods. In this sense, KET and TopoCoend share the same categorical ambition: both define Kan- or coend-style aggregation operators, but over different types of learned or prescribed structure. Whether a particular operator has the advertised universal property is a separate theorem obligation.

Empirical Questions

The experiments in this chapter are guided by four main questions:

1. Can structured Kan-style aggregation improve strictly causal language modeling relative to ordinary Transformers and GT?
2. How much of the gain comes from architecture, and how much comes from the information regime?
3. When is quadratic global aggregation worth its additional computational cost, and when is incidence restriction sufficient?
4. Can learned latent-space neighborhoods in TopoCoend provide a competitive alternative to fixed simplicial structure?

We answer these questions on Penn Treebank, WikiText-2, and WikiText-103, comparing perplexity, runtime, and hardware sensitivity across strict-causal, gold-noncausal, and predict-detach regimes.

CONTRIBUTIONS. At a high level, this chapter makes five contributions:

1. **Universal-extension perspective:** it reframes attention and geometric mixing as structured extension operators, rather than as unrelated architectural heuristics.
2. **KET architecture:** it develops quadratic and incidence-restricted Kan Extension Transformers and compares GT with an efficient incidence-restricted member of the same design family.
3. **TopoCoend architecture:** it develops a coend-motivated alternative in which neighborhoods arise from learned low-dimensional geometry.
4. **Causality diagnosis:** it sharply separates valid and invalid information regimes, showing that gold noncausal mixing leaks while predict-and-detach blocks an auxiliary gradient path; strict autoregression additionally requires the target-relative filtration condition.
5. **Empirical evidence:** it reports perplexity–runtime tradeoffs on PTB, WikiText-2, and WikiText-103, showing that information regime often dominates architectural detail.

Related Work

Self-conditioning and iterative refinement in language modeling have appeared under several names, including prompt repetition, two-pass decoding, and self-consistency mechanisms. Predict-and-detach provides an operational comparison with these phenomena, but it is not equivalent to a second autoregressive pass. A valid second pass conditions on already generated context; the reported future-carrier runs instead read carriers computed at future teacher-forced positions. The two must not be conflated.

From the categorical machine learning side, a growing body of work has interpreted neural architectures and backpropagation in terms of functors, compositional structure, and universal constructions. The present chapter builds directly on that line of work. Its distinctive move is to interpret both attention and geometric mixing as enriched extension operators over simplicial or topological neighborhood systems in the setting of autoregressive language modeling. This yields not only new architectures, but also a sharper diagnosis of when structured mixing is valid, when it leaks, and why self-conditioned nonlocal aggregation can work at all.

Kan Extensions as Universal Structured Extension

A recurring problem in machine learning is the following: we are given information on a small or local domain, and we wish to extend it to a larger structured domain. For ordinary functions on sets, this problem is typically non-canonical. If values are known only on finitely many samples, there are usually many incompatible ways to extend them: constant predictors, linear

interpolants, splines, kernels, high-degree polynomials, and so on. Nothing in the bare set-theoretic formulation singles out one extension as the correct one.

Category theory changes the formulation of the problem. Instead of asking for an arbitrary extension of a function on points, we ask for an extension of a functor along a structural map. Once the source values and the target structure are organized functorially, universal constructions become available. The most important of these for the present chapter are *left* and *right Kan extensions*. They provide principled ways to extend local information across a larger domain relative to a specified structure.

This perspective is central to the architectures developed in this chapter. A Transformer layer, a geometric mixing layer, or a higher-order message-passing rule may all be viewed as structured extension operators. The key question is not merely *which values are mixed*, but *along what structure* and according to *which universal principle* they are extended.

Informal Definition

Let

$$F : \mathcal{C} \rightarrow \mathcal{E}, \quad K : \mathcal{C} \rightarrow \mathcal{D}$$

be functors. The functor F assigns values to objects of the source category $\mathcal{C}$, while K explains how $\mathcal{C}$ sits inside, or maps into, the larger target category $\mathcal{D}$. We would like to construct a functor

$$H : \mathcal{D} \rightarrow \mathcal{E}$$

that extends F along K .

Definition 44. *The **left Kan extension** of F along K , denoted $\text{Lan}_K F$, is a functor*

$$\text{Lan}_K F : \mathcal{D} \rightarrow \mathcal{E}$$

together with a natural transformation

$$\eta : F \Rightarrow (\text{Lan}_K F) \circ K$$

such that for any other pair

$$G : \mathcal{D} \rightarrow \mathcal{E}, \quad \gamma : F \Rightarrow G \circ K,$$

there exists a unique natural transformation

$$\alpha : \text{Lan}_K F \Rightarrow G$$

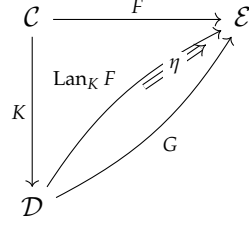
with

$$\gamma = (\alpha K) \circ \eta.$$

*Dually, the **right Kan extension** $\text{Ran}_K F$ is a functor equipped with*

$$\epsilon : (\text{Ran}_K F) \circ K \Rightarrow F$$

such that for every $G : \mathcal{D} \rightarrow \mathcal{E}$ and $\gamma : G \circ K \Rightarrow F$, there is a unique $\alpha : G \Rightarrow \text{Ran}_K F$ satisfying $\gamma = \epsilon \circ (\alpha K)$.



The slogan is simple:

A left Kan extension is the most universal way to *build* values on $\mathcal{D}$ from data on $\mathcal{C}$. A right Kan extension is the universal object representing compatible cones from values on $\mathcal{C}$, when that extension exists.

For the purposes of this chapter, the left Kan extension will play the leading role, since it captures aggregation and contextualization. The right Kan point of view will become increasingly important when we turn to denoising, completion, and refinement.

Pointwise Formulas: Colimits and Limits

Kan extensions become especially intuitive when written pointwise. For an object $d \in \mathcal{D}$, the value $(\text{Lan}_K F)(d)$ is built from all source objects $c \in \mathcal{C}$ that map into d . Formally, one considers the comma category $(K \downarrow d)$, whose objects are arrows

$$Kc \rightarrow d$$

in $\mathcal{D}$, and whose morphisms are commuting triangles induced by morphisms in $\mathcal{C}$.

If the relevant colimits exist, then the left Kan extension is computed pointwise by

$$(\text{Lan}_K F)(d) \cong \text{colim}_{(K \downarrow d)} F. \quad (2)$$

Dually, the right Kan extension is built from all ways in which d must be compatible with source objects. Writing $(d \downarrow K)$ for the comma category whose objects are arrows

$$d \rightarrow Kc,$$

one obtains the pointwise formula

$$(\text{Ran}_K F)(d) \cong \lim_{(d \downarrow K)} F. \quad (3)$$

These two formulas already capture the conceptual distinction that will matter throughout the chapter:

- Lan uses a **colimit**: it glues together many local contributions.

- Ran uses a **limit**: it finds a jointly compatible value satisfying many constraints.

For machine learning readers, it is often enough to remember the mnemonic:

left Kan = colimit-like extension from specified sources, right Kan =
limit-like compatible extension.

Interpolation and Completion as Motivating Examples

The difference between sets and categories can be seen in two familiar problems.

INTERPOLATION. Suppose we know values $f(x_i) = y_i$ at a finite set of sample points. As a set-theoretic problem, there is no canonical extension to all query points x : one may choose a constant predictor, a piecewise linear interpolant, a polynomial, or a kernel smoother. The ambiguity does not disappear in category theory, but it is reframed. One specifies a structural relation $W(i, x)$ saying how each sample x_i contributes to a query point x , and the extension is then computed relative to that structure. In finite enriched settings, the resulting left Kan extension takes the weighted form

$$(\text{Lan}_W F)(x) \approx \sum_i W(i, x) F(i).$$

Thus the Kan extension is not a magical elimination of modeling choices; it is the universal extension *relative to the chosen structure*.

COMPLETION AND DENOISING. Now suppose the problem is not to interpolate from samples but to complete a partially observed or noisy state. Again, many completions may be possible in principle. What selects among them is additional structure: a noise model, objective, prior, or exact compatibility declaration. A pointwise right Kan extension computes a limit of a specified diagram; it does not by itself choose an approximate denoising solution or define a diffusion process. It can serve as an exact comparison target when completion really is a compatible-cone problem, while learned denoising requires further statistical semantics.

These two examples foreshadow the main architectural claim of the chapter: attention has the form of a learned weighted coend and becomes an actual enriched left Kan extension only under the representability and functoriality conditions stated above, while some exact structured-completion problems may admit a right-Kan interpretation once their diagram and compatibility maps are supplied.

Coends as Weighted Aggregation

The pointwise left Kan formula in Eq. (2) can often be written in a more computationally useful form using a *coend*. For readers unfamiliar with ends

and coends, the essential intuition is simple: a coend behaves like a structured weighted sum, with the indexing category enforcing the appropriate identifications.

The notation

$$\int^c H(c, c)$$

should be read informally as

“sum over all c , while identifying contributions that are equivalent under the morphisms of the indexing category.”

In the settings relevant here, this behaves like a weighted pooling operation. If $F(c)$ is a feature vector and $W(c, t)$ is a compatibility weight, then the coend has the computational flavor

$$(\text{coend})(t) \sim \sum_c W(c, t) F(c),$$

possibly with identifications imposed by the structure of the category.

This is exactly the pattern that appears in attention, geometric message passing, and Kan-style pooling: values are aggregated from many structured sources, and the weights encode how strongly each source contributes to the target.

Kan Extensions via Coends

A standard enriched formula expresses the pointwise left Kan extension as a coend:

$$(\text{Lan}_K F)(d) \cong \int^{c \in \mathcal{C}} \mathcal{D}(Kc, d) \otimes F(c). \quad (4)$$

Here $\mathcal{D}(Kc, d)$ is the enriched hom-object and $\otimes$ is the tensoring of the target category over the enriching category. Ordinary scalar weighting of feature vectors is one possible realization only after an appropriate enrichment and action have been specified. Equation (4) therefore says:

To compute the value at d , gather contributions from every source object c , weight them by how strongly Kc points to d , and form the coend quotient prescribed by the enriched category.

Dually, in enriched settings the right Kan extension may be written in end form as

$$(\text{Ran}_K F)(d) \cong \int_{c \in \mathcal{C}} [\mathcal{D}(d, Kc), F(c)], \quad (5)$$

where the end expresses compatibility with all incoming constraints. We will not need this formula computationally in most of the chapter, but it is conceptually important: left Kan extensions are naturally coend-like and aggregation-oriented, while right Kan extensions are end-like and compatibility-oriented.

Interpretation for Transformer Architectures

Equation (4) has a direct neural interpretation.

- The source objects c correspond to tokens, edges, or higher-order simplices.
- The target object d is typically a token position whose representation is being updated.
- The hom-objects $\mathcal{D}(Kc, d)$ play the role of compatibility or neighborhood weights.
- The values $F(c)$ are the feature vectors carried by the source objects.

Under this interpretation, a familiar update

$$h'_d = \sum_c w_{d,c} V_c$$

has the computational form of a discretized weighted coend. It is precisely an enriched Kan extension only after specifying an enrichment for which the weights represent the relevant hom-objects and the update satisfies the associated universal property. Generic learned weights do not establish that claim by themselves. In particular, query–key weights are input-dependent and normalized separately at each target. That is a useful learned kernel, but softmax normalization alone proves neither enriched composition of hom-objects nor functoriality in the source and target variables. The architectural families in this chapter can then be read in a common language:

- **Transformers** aggregate over token neighborhoods; attention is a weighted left-Kan-style pooling rule on a sequence.
- **Kan Extension Transformers (KET)** make the source category explicit, lifting from tokens to simplices and then extending back to token updates via simplicial incidence.
- **TopoCoend Transformers** retain the weighted coend form but define the neighborhood category through learned latent metric geometry rather than fixed simplicial structure.

Thus KET and TopoCoend should not be viewed as unrelated alternatives. They are two realizations of the same broader design idea: structured contextualization compared against universal weighted extension. The generic learned operators remain Kan- or coend-style rather than certified universal constructions.

Why This Matters for the Chapter

This section provides the conceptual bridge from ordinary geometric mixing to the models introduced later.

- It explains why attention and geometric aggregation can be treated uniformly as structured extension operators.
- It clarifies why KET naturally lives in the language of Kan extensions: the model extends values from simplicial structure back to tokens.
- It explains why TopoCoend lives naturally in the language of coends: the model aggregates over learned neighborhoods using weighted geometric compatibility.
- It prepares the later analysis of information regimes by distinguishing aggregation from compatibility and extension from leakage.

The main lesson is methodological. Kan extensions and coends supply a precise universal vocabulary against which a learned aggregation can be tested. They also make clear what a weighted-sum implementation has not yet established: functorial source values, categorical weights, the coend identifications, and the relevant universal property.

Neighborhood Systems and Kan-Style Aggregation

The model families studied in this chapter differ less in the form of their update rule than in the *structure of the neighborhoods* over which they aggregate. Standard attention aggregates over tokens. TopoCoend aggregates over learned latent-space neighborhoods. Kan Extension Transformers aggregate over simplicial objects such as tokens, edges, and higher-order motifs. This section gives a common categorical language for these choices.

A Common Setup

Let

$$\mathcal{T} = \{0, \dots, S-1\}$$

denote the target token positions in a sequence window of length S . A model update at token t is built from a *source neighborhood system* $\mathcal{N}$, whose objects may be tokens, edges, simplices, or learned geometric neighbors.

At the categorical target level we would write

$$X : \mathcal{N} \rightarrow \mathbf{Vect}$$

for a functor assigning feature spaces and linear restriction or transport maps to source objects and morphisms. The implementations below construct feature tensors on a finite list of objects; unless their morphism maps and

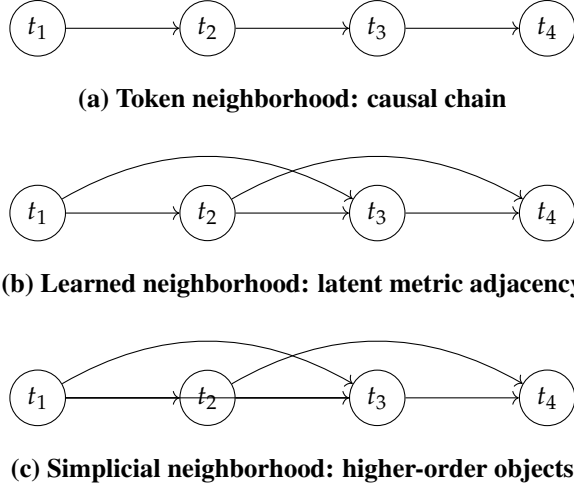


Figure 25: Three neighborhood systems underlying the architectures of this chapter. (a) Standard attention uses token-level neighborhoods. (b) TopoCoend adds learned edges induced by geometric proximity in a low-dimensional latent space. (c) KET enlarges the source structure to include higher-order simplices, so aggregation may occur over edges, faces, and motifs rather than only tokens.

functor laws are registered and verified, they realize the object-level part of this target rather than a full functor. For example, $X(\sigma)$ may be a token state, an edge embedding, or a higher-order simplex value. The contribution of a source object $\sigma \in \mathcal{N}$ to a target token $t \in \mathcal{T}$ is determined by a structural weight

$$W(t, \sigma),$$

which may arise from attention scores, latent-space kernels, or simplicial incidence.

This yields a common weighted aggregation form

$$h'_t \approx \int^{\sigma \in \mathcal{N}} W(t, \sigma) \otimes X(\sigma) \approx \sum_{\sigma \in \mathcal{N}} w(t, \sigma) V(\sigma), \quad (6)$$

where the first expression is the categorical target and the second is its finite weighted-sum analogue. The approximation sign is substantive: a learned sum does not automatically impose the dinaturality quotient of a coend.

Equation (6) is the computational form shared by attention, TopoCoend, and KET. What changes from one model family to another is the choice of $\mathcal{N}$ and the origin of the weights $W(t, \sigma)$.

Token Neighborhoods: Attention

For standard self-attention, the source neighborhood system contains only token objects:

$$\mathcal{N}_{\text{attn}} = \{0, \dots, S-1\}.$$

Each source object is a token s , and the update at target token t takes the familiar form

$$h'_t = \sum_s w(t, s) V_s.$$

In the categorical language of the previous section, this is a weighted left-Kan-style extension in which the source and target are both token positions. Thus attention is already a structured extension operator; KET and TopoCoend extend this computational template by changing the source neighborhood system. An exact generalization claim would require explicit comparison functors and a proof that the associated universal constructions are preserved.

Learned Latent Neighborhoods: TopoCoend

TopoCoend keeps the source objects at the token level but changes the neighborhood relation. Instead of using only positional adjacency or dense token-to-token attention, it learns a low-dimensional geometric representation

$$z_t = \pi(v_t),$$

where v_t is either a hidden state or a detached predictive carrier. A weighted k -nearest-neighbor graph is then constructed in the latent space of the z_t 's.

The resulting neighborhood system $\mathcal{N}_{\text{topo}}$ is therefore a learned geometric graph. Its weights

$$W_{\text{topo}}(t, s)$$

are induced by latent metric proximity rather than by sequence position alone. The corresponding update is

$$h'_t = h_t + \sum_s w_{\text{topo}}(t, s) V_s. \quad (7)$$

TopoCoend is thus a coend-style weighted aggregation over a learned neighborhood graph. Calling that graph a category requires identities and compositional closure; calling the update a coend additionally requires the associated identifications.

The adjective *topological* is therefore an architectural name, not a proved invariance property. The implementation learns coordinates, computes metric nearest neighbors, and applies normalized weights. It does not establish continuity under changes of coordinates, compute a topological invariant, construct a UMAP-style fuzzy simplicial set, or verify a coend universal property.

Simplicial Neighborhoods: KET

KET changes not just the weights but the very objects over which aggregation is performed. The source neighborhood system now includes simplicial objects:

- 0-simplices: tokens t ,
- 1-simplices: edges such as $(t - 1, t)$,

- optional higher simplices: faces, motifs, or larger spans.

Let

$$\mathcal{N}_{\text{simp}}$$

denote this simplicial indexing category. Each simplex σ is assigned a value $X(\sigma)$, and the contribution of σ to token t is controlled by a simplicial weight

$$W_{\text{simp}}(t, \sigma).$$

The generic KET update is therefore

$$h'_t = h_t + \sum_{\sigma \in \mathcal{N}_{\text{simp}}} w_{\text{simp}}(t, \sigma) V(\sigma). \quad (8)$$

The key point is that KET allows aggregation over *higher-order source objects*, not just over individual tokens. This enlarges the implementation’s carrier vocabulary beyond individual tokens and provides a testable generalization of attention-like and local geometric mixing.

Quadratic and Incidence-Restricted KET

The simplicial update in Eq. (8) admits two important realizations.

QUADRATIC KET. In the global or quadratic variant, every token may aggregate from every simplex:

$$h'_t = h_t + \sum_{\sigma \in \mathcal{N}_{\text{simp}}} w(t, \sigma) V(\sigma), \quad w(t, \sigma) = \text{softmax}(Q_t^\top K_\sigma). \quad (9)$$

This is a second attention-like kernel, now defined over simplices rather than only over tokens. If the simplex list has $O(S)$ objects, as in the edge-only implementation, the token–simplex interaction has complexity $O(S^2)$. Including all higher-dimensional simplices can enlarge the list and therefore increase this cost; “quadratic” describes the present edge-level construction rather than an arbitrary full simplicial complex.

INCIDENCE-RESTRICTED KET. In the incidence-restricted variant, only simplices incident to a token may contribute:

$$h'_t \approx h_t + \sum_{\sigma \ni t} \phi(V(\sigma)). \quad (10)$$

For edge-only models this becomes

$$e_t = \psi([v_{t-1}, v_t]), \quad (11)$$

$$h'_t = h_t + \phi(e_t) \quad (\text{causal incidence}). \quad (12)$$

This reduces the complexity to $O(S)$ and makes the connection to GT transparent.

Relationship to Geometric Transformers

The geometric branch of a Geometric Transformer can be compared with an incidence-restricted Kan-style update in which the message map is implemented by a local convolutional or message-passing operator. This yields a design table, not a proven inclusion hierarchy:

- **Attention:** weighted extension over tokens.
- **TopoCoend:** weighted extension over learned latent-space neighborhoods.
- **Incidence-restricted KET:** weighted extension over incident simplices.
- **Quadratic KET:** weighted extension over all simplices with a learned global kernel.
- **GT:** a local operator comparable to an efficient incidence-restricted member of the KET design family.

This comparison organizes the runtime and inductive-bias tradeoffs observed later in the chapter; it does not establish strict expressivity inclusions.

Causality and Information Regimes

Autoregressive language models must respect the factorization

$$p(x_{t+1} \mid x_{\leq t}),$$

so every computation contributing to the logits at position t must depend only on the prefix $x_{\leq t}$. Once we enlarge the neighborhood system beyond the causal chain, the critical question is no longer merely *which neighborhood objects are available*, but *how their values are constructed*.

Three Information Regimes

We distinguish three regimes throughout the chapter:

- **Strict causal:** only prefix information contributes to the update at t .
- **Gold noncausal (invalid):** future-position hidden states built under teacher forcing are allowed to contribute to the update at t .
- **Predict-and-detach (noncausal self-conditioned in the reported future-carrier runs):** future-position carriers are permitted structurally, but their values are derived from prefix-conditioned predictions and detached from gradient flow.

The distinction between the second and third regimes is central. They may use similar neighborhood structure, but detachment alone does not make the latter strictly autoregressive. A future carrier at $s > t$ computed from $x_{\leq s}$ violates the filtration required at target t .

Causality Lives in the Values

A Kan-style update has the generic form

$$h'_t \approx \sum_{\sigma \in \mathcal{N}} w(t, \sigma) V(\sigma).$$

Whether this is autoregressively valid depends on the construction of $V(\sigma)$.

If $V(\sigma)$ is built from hidden states that already encode gold future tokens under teacher forcing, then allowing a future-containing simplex or neighbor to contribute to t leaks the answer. If, by contrast, every $V(\sigma)$ consumed at target t is built only from $x_{\leq t}$, then the same structural neighborhood can be used without future leakage. “Predictive” and “detached” do not establish this target-relative condition.

Gold Noncausal Regime (Invalid)

Suppose $V(\sigma)$ is constructed from hidden states h_s computed under teacher forcing, and suppose some σ involving a future position $s > t$ contributes to the update at t . Then $V(\sigma)$ may encode gold token x_s , which can flow into the logits used to predict the very same future token.

Proposition 1 (Future-token leakage). *If future-position hidden states built from teacher-forced gold tokens contribute noncausally to the update at position t , then the model can reduce perplexity by directly exploiting gold future information.*

Sketch. Under teacher forcing, h_s is a deterministic function of $x_{\leq s}$, including the gold token x_s . If h_s or a simplex value derived from it contributes to the update at $t < s$, then the logits at t can encode information about a future gold token. This invalidates the autoregressive factorization by making future information available. Whether optimization exploits the path is empirical; the recorded direct-future controls do so and approach unit perplexity. $\square$

Strict Causal Regime

The safest regime is to restrict both the neighborhood structure and the value construction so that only prefix objects may contribute:

$$\max(\sigma) \leq t.$$

Together with prefix-only construction of every simplex value and weight, this preserves autoregressive validity but may sacrifice some of the benefit of richer structured neighborhoods.

Predict-and-Detach Self-Conditioning

One proposed carrier construction replaces hidden-state values by detached predictive embeddings. Let

$$\ell_t = W_o h_t$$

be the causal logits at position t . Define the predictive embedding

$$\hat{e}_t = \text{detach}\left(\text{softmax}(\ell_t/T) E\right), \quad (13)$$

where E is the embedding matrix and T is a temperature.

The carrier $\hat{e}_t$ has the same forward provenance as h_t . Detachment only removes the auxiliary reverse-mode derivative path.

Proposition 2 (Target-relative autoregressive criterion). *Consider an update at target position t . If every carrier $c_{s \rightarrow t}$ consumed by that update is measurable with respect to $\sigma(x_{\leq t})$, and the remaining update is causal, then the resulting representation at t is a function of the target prefix alone.*

Sketch. Measurability is preserved under composition with the causal update. Consequently the updated representation, and hence its output logits, remain $\sigma(x_{\leq t})$ -measurable. Detachment is not part of this forward measurability argument: it changes derivatives, not the information available to the forward computation. $\square$

In particular, $\hat{e}_s$ is measurable with respect to $\sigma(x_{\leq s})$. When $s > t$, this does not imply measurability with respect to $\sigma(x_{\leq t})$. Future-indexed predict-detach carriers used in the experiments therefore constitute noncausal self-conditioning unless they are recomputed from the target prefix.

Prompt Repetition as an Internal Mechanism

Predict-and-detach is superficially reminiscent of prompt repetition, but the provenance protocols differ. In external prompt repetition, generated tokens are fed back into the context for a second pass. Here, predictive carriers derived from the model’s own prefix-based beliefs are fed into a structured aggregation mechanism within the same overall model. A genuinely autoregressive KET or TopoCoend second pass could consume only previously generated carriers. The reported simultaneous future-carrier runs do not implement that protocol and are therefore retained as Regime-A diagnostics.

Model Instantiations

We now specialize the general framework to the concrete model families studied experimentally.

Backbone Transformer

All models share a causal Transformer backbone. Given token embeddings and positional embeddings, define

$$h_t^{(0)} = \text{Embed}(x_t) + p_t.$$

Each layer applies masked self-attention and a feedforward sublayer, yielding final hidden states h_t and logits

$$\ell_t = W_o h_t.$$

By construction, the backbone is strictly causal: h_t depends only on $x_{\leq t}$.

Quadratic KET

Quadratic KET augments the backbone with a global simplicial aggregation branch. Let $\mathcal{N}_{\text{simp}}$ contain tokens, adjacent edges, and optionally higher simplices. Each simplex σ is assigned a value $V(\sigma)$, built either from hidden states or from detached predictive carriers.

The update is

$$h'_t = h_t + \sum_{\sigma \in \mathcal{N}_{\text{simp}}} w(t, \sigma) V(\sigma), \quad w(t, \sigma) = \text{softmax}(Q_t^\top K_\sigma). \quad (14)$$

A simplicial causal mask may be imposed by requiring

$$\max(\sigma) \leq t.$$

Without such a mask, validity depends on the target-relative provenance of every value $V(\sigma)$; predict-and-detach alone is insufficient.

Incidence-Restricted KET

The incidence-restricted model uses only simplices incident to each token. For edge-only neighborhoods:

$$e_t = \psi([v_{t-1}, v_t]),$$

where v_t is either a hidden state or a detached predictive carrier. Token updates are then

$$h'_t = h_t + \phi(e_t) \quad (\text{causal incidence}), \quad (15)$$

$$h'_t = h_t + \phi(e_t) + \phi(e_{t+1}) \quad (\text{noncausal incidence}). \quad (16)$$

This is the linear-time KET approximation studied throughout the chapter.

TopoCoend Transformer

TopoCoend replaces simplicial source objects by a learned geometric neighborhood over token positions. Given carriers v_t , compute latent coordinates

$$z_t = \pi(v_t),$$

construct a weighted k -nearest-neighbor graph in the z -space, and use the resulting weights $w_{\text{topo}}(t, s)$ in the update

$$h'_t = h_t + \sum_s w_{\text{topo}}(t, s) V_s.$$

As with KET, the strict-causal and predict-and-detach regimes differ in how the carriers v_i are constructed and in which target-relative information those carriers expose.

GT as a Local Comparison Case

Geometric Transformers fit naturally into this framework. Their geometric branch is an incidence-restricted Kan-style update in which the message maps are fixed to local convolutional or graph-style operators on the one-dimensional token chain. At the level of computational templates:

- GT supplies an efficient local operator,
- incidence-restricted KET learns an edge-incidence message map,
- quadratic KET learns a global simplex kernel,
- TopoCoend replaces simplicial neighborhoods by learned latent-space ones.

Summary of the Hierarchy

The model families can be ordered by the neighborhoods exposed to their learned aggregators—tokens, local edges, incident simplices, or all listed simplices—but this does not prove inclusion of their represented function classes. TopoCoend provides a complementary branch in which the source neighborhood graph is learned geometrically rather than fixed simplicially.

This hierarchy will guide the empirical comparisons that follow.

Predict–Detach as Paired Operational Semantics

The predict–detach mechanism appears in several experimental variants in this chapter. Operationally, it is simple: the model computes a predictive carrier and then applies `detach()` so that the carrier may participate in forward computation without transmitting gradients backward through the predictive branch. Conceptually, it exposes a separation between two kinds of structure: the *forward semantics* of the model and the *learning semantics* induced by reverse-mode differentiation.

This section gives a paired operational description of that separation. It does not claim that framework-level `detach()` is intrinsically a modality in a specified differential category, nor that zeroing a derivative makes a noncausal forward computation valid.

Forward and Backward Semantics

A neural layer has two mathematically distinct aspects.

FORWARD SEMANTICS. A forward operator transforms representational states:

$$f : X \rightarrow Y,$$

where X and Y may be token sequences, hidden states, predictive distributions, or embedding carriers.

BACKWARD SEMANTICS. Under reverse-mode automatic differentiation, the same operator induces a backward map on cotangent or gradient objects:

$$f^* : \partial Y \rightarrow \partial X.$$

Thus every learned computation carries both a forward action and a backward action. For the purposes of this chapter, it is convenient to package these together into a paired semantics

$$f \mapsto (f, f^*).$$

Detach as a Stop-Gradient Transformation

The detach operation preserves the forward component of an operator but kills its backward component.

Definition 45 (Paired stop-gradient semantics). *For an operator with paired semantics (f, f^*) , define*

$$\square(f, f^*) := (f, 0).$$

Equivalently, $\square$ leaves the forward map unchanged while replacing the backward map by the zero morphism.

This is an operational definition on registered forward/backward pairs. To promote $\square$ to a categorical modality would require a category of such pairs together with functoriality and the relevant modal laws.

This formalizes the implementation-level meaning of `detach()`: the carrier remains available to subsequent layers as a forward signal, but it cannot serve as a path for gradient propagation.

INTERPRETATION. The stop-gradient transformation $\square$ is not a change to what the model computes in the forward pass. It is a change to which paths are visible to learning. In this operational sense, predict–detach imposes a boundary between *semantic use* and *gradient credit assignment*.

The Predict–Detach Pair

Let

$$\text{Pred} : H \rightarrow E$$

denote the predictive carrier map from hidden states H to embedding space E :

$$\text{Pred}(h) = \text{softmax}(Wh/T) E.$$

The paired semantics of the predict–detach carrier is

$$\text{PredDetach} = \square(\text{Pred}, \text{Pred}^*).$$

Thus a hidden state first generates a predictive embedding and that embedding is then passed into later geometric or simplicial aggregation only through the detached channel:

$$H \xrightarrow{\text{Pred}} E \quad \text{with the auxiliary backward map replaced by zero.}$$

In paired semantics, this becomes

$$(\text{Pred}, \text{Pred}^*) \mapsto (\text{Pred}, 0).$$

So the predictive carrier participates in forward aggregation, but the predictive branch itself does not receive gradient contributions from that later aggregation.

Information Regimes Revisited

This paired viewpoint clarifies the three information regimes used throughout the chapter.

STRICT CAUSAL REGIME. All forward morphisms respect temporal order. No future-position object contributes to the update at time t . This is the ordinary autoregressive regime.

GOLD NONCAUSAL REGIME (INVALID). Future-position hidden states computed under teacher forcing are allowed to contribute directly to the update at time t . Since those hidden states encode gold future tokens, this introduces an illegitimate information path.

PREDICT–DETACH REGIME. Future-position carriers may appear structurally in the neighborhood system, but only through predictive values passed through stop-gradient. Thus the forward computation may exploit a richer structure while the auxiliary branch receives no gradient through that channel. If a future carrier is consumed, the forward computation is still non-causal.

Two distinctions are required: whether the forward provenance is causal, and whether the auxiliary derivative path is active. Hidden versus predicted carriers can change the severity of leakage, but neither carrier type excuses a target-relative provenance violation.

What Predict–Detach Does and Does Not Guarantee

Predict–detach separates two questions that should not be conflated. The detach operation blocks backward influence through an auxiliary carrier, while autoregressive validity is a property of the forward information flow.

Let

$$\hat{e}_t = \text{detach}\left(\text{softmax}(\ell_t/T) E\right), \quad \ell_t = W_o h_t.$$

Since h_t is causal, $\hat{e}_t$ is a function of $x_{\leq t}$. It can safely update target t , or a later target, provided the target-relative filtration criterion above is satisfied. It cannot in general update an earlier target $u < t$ while retaining strict autoregressive semantics, because $x_{\leq t}$ contains information beyond $x_{\leq u}$.

Gold hidden-state carriers and future-indexed predicted carriers both fail the target-relative test in the reported simultaneous protocol. Predict–detach changes the representation transported and blocks its auxiliary gradient path, but a carrier computed at a future teacher-forced position still gives an earlier target noncausal forward information.

Diagrammatic View

The contrast between an ordinary predictive branch and a detached one can be summarized diagrammatically as

$$\begin{array}{ccc} H & \xrightarrow{\text{Pred}} & E \\ \downarrow & & \downarrow \\ \partial H & \xleftarrow{\text{Pred}^*} & \partial E \end{array} \quad \rightsquigarrow \quad \begin{array}{ccc} H & \xrightarrow{\text{PredDetach}} & E \\ \downarrow & & \downarrow \\ \partial H & \xleftarrow{0} & \partial E \end{array}$$

The left diagram represents an ordinary predictive carrier with both forward and backward semantics. The right diagram represents the detached carrier: the forward morphism remains, while the backward morphism is zeroed out.

Relation to KET and TopoCoend

This paired interpretation applies uniformly across the models of the chapter.

- In **quadratic KET**, detached predictive carriers may be lifted to simplex values and then aggregated globally.
- In **incidence-restricted KET**, detached carriers may be passed along local simplicial incidence relations.
- In **TopoCoend**, detached carriers may be embedded into a learned coordinate system and used to define weighted nearest-neighbor neighborhood aggregation.

Thus predict–detach is not specific to one architecture. It is a general device for separating structural forward computation from auxiliary gradient credit assignment. Causal validity must be certified independently.

Conceptual Summary

The main point of this section is that predict–detach should not be viewed merely as a framework-specific trick. It has a useful paired operational meaning:

predict–detach preserves a forward carrier while replacing one registered backward map by zero.

This perspective explains what the device accomplishes uniformly across GT, KET, and TopoCoend architectures without labeling all such variants strictly autoregressive.

Datasets, Protocol, and Experimental Setup

We evaluate the model families introduced above on Penn Treebank (PTB), WikiText-2 (WT2), and WikiText-103 (WT103), all using standard word-level tokenization consistent with prior language-modeling work. The headline language-model comparisons use a causal Transformer backbone with $L = 2$ layers, hidden dimension $d = 256$, 4 attention heads, and context length 128. The tables in this chapter report one run per configuration at this headline setting. They are not a multi-seed or factorial architecture study.

Models are trained for 5000 iterations using AdamW with learning rate 3×10^{-4} . We report validation and test perplexity (PPL), with lower values indicating better predictive performance. For runtime comparisons, we measure end-to-end wall-clock throughput in iterations per second (it/s), including periodic evaluation. Experiments were run on both Apple MPS hardware and an NVIDIA Blackwell-based DGX Spark GPU; unless explicitly stated otherwise, the main comparison tables report Blackwell GPU results.

Model Families

We compare the following model classes.

TRANSFORMER (AR-BASE). A standard causal Transformer with masked self-attention and no additional geometric or simplicial branch.

GEOMETRIC TRANSFORMER (GT). A Transformer augmented with a geometric mixing branch. We consider:

- **GT-CausalGeom:** strict-causal geometric convolution.
- **GT-CausalSelfHint:** causal geometric mixing with detached predictive self-hint.

QUADRATIC KAN TRANSFORMER (KAN-QUAD). A global simplicial aggregation layer with a learned kernel over tokens and edges (and optionally higher simplices). We evaluate:

- **Kan-Quad-C**: strict simplicial causal masking.
- **Kan-Quad-A(P)**: noncausal simplicial structure with future-indexed detached predictive carriers.

INCIDENCE-RESTRICTED KAN TRANSFORMER (KAN-INC). A linear-time edge-based or incidence-based approximation to full simplicial Kan pooling. We evaluate:

- **Kan-Inc-C**: causal incidence restriction.
- **Kan-Inc-A(P)**: noncausal incidence with future-indexed detached predictive carriers.

TOPOCOEND TRANSFORMER (TOPOCOEND). A model in which token states are projected into a learned low-dimensional coordinate space, a weighted k -nearest-neighbor graph is constructed, and aggregation is performed over these learned neighborhoods. We evaluate:

- **TopoCoend-C[†]**: intended-causal latent-neighborhood aggregation. This implementation has an empty-neighborhood defect at $t = 0$, described below, so this row is not included among certified strict-causal baselines.
- **TopoCoend-A(P)**: noncausal latent-space neighborhoods with future-indexed detached predictive carriers.

INVALID GOLD-NONCAUSAL DIAGNOSTICS. For diagnosis only, we also include regimes that explicitly mix hidden states encoding gold future tokens. These models collapse perplexity toward 1 and are excluded from fair autoregressive comparison.

Information Regimes

All experiments are organized by the three information regimes developed earlier in the chapter:

1. **Strict causal**: all updates depend only on prefix-derived states or simplices satisfying the causal mask.
2. **Gold noncausal (invalid)**: future-position hidden states built under teacher forcing contribute directly.
3. **Future predicted-carrier A(P)**: noncausal structure uses detached predicted carriers computed at future teacher-forced positions.

This distinction is central to the experimental narrative. The strict-causal regime gives the fairest comparison of architectural expressivity under autoregressive validity. The A(P) regime tests how readily each architecture exploits a subtler future-informed carrier; it is not an autoregressive comparison. The gold-noncausal regime is retained only to confirm the leakage analysis.

Learning Dynamics

PTB

The figure below shows validation loss and validation perplexity on PTB.

STRICT-CAUSAL REGIME. On PTB, the standard autoregressive baseline remains strongest among the strictly causal models. GT-CausalGeom and GT-CausalSelfHint track the baseline closely. Quadratic KET converges somewhat more slowly and reaches a slightly higher final perplexity, while incidence-restricted KET trades modest accuracy for improved throughput. This is consistent with the view that on smaller corpora the additional expressivity of global simplicial pooling is not yet decisive.

FUTURE PREDICTED-CARRIER DIAGNOSTIC. Allowing noncausal structure through future-indexed detached predictive carriers dramatically changes the teacher-forced metric. Both Kan-Inc-A(P) and Kan-Quad-A(P) obtain much smaller perplexities than their strict-causal counterparts. By contrast, gold-noncausal regimes collapse toward perplexity near 1, confirming that hidden-state noncausal mixing leaks gold future tokens.

INTERPRETATION. PTB already exhibits the chapter’s central diagnostic lesson: forward provenance dominates the numerical comparison. Detachment does not make the A(P) curves valid autoregressive results.

WikiText-2

The figure below shows learning dynamics on WikiText-2.

STRICT-CAUSAL REGIME. On WikiText-2, the separation among strict-causal architectures becomes clearer than on PTB. GT improves over the autoregressive baseline, and Quadratic KET achieves the best strict-causal perplexity among the models considered. Incidence-restricted KET remains competitive while offering improved throughput relative to full global simplicial pooling.

FUTURE PREDICTED-CARRIER DIAGNOSTIC. As on PTB, the A(P) variants obtain order-of-magnitude lower teacher-forced perplexity than the

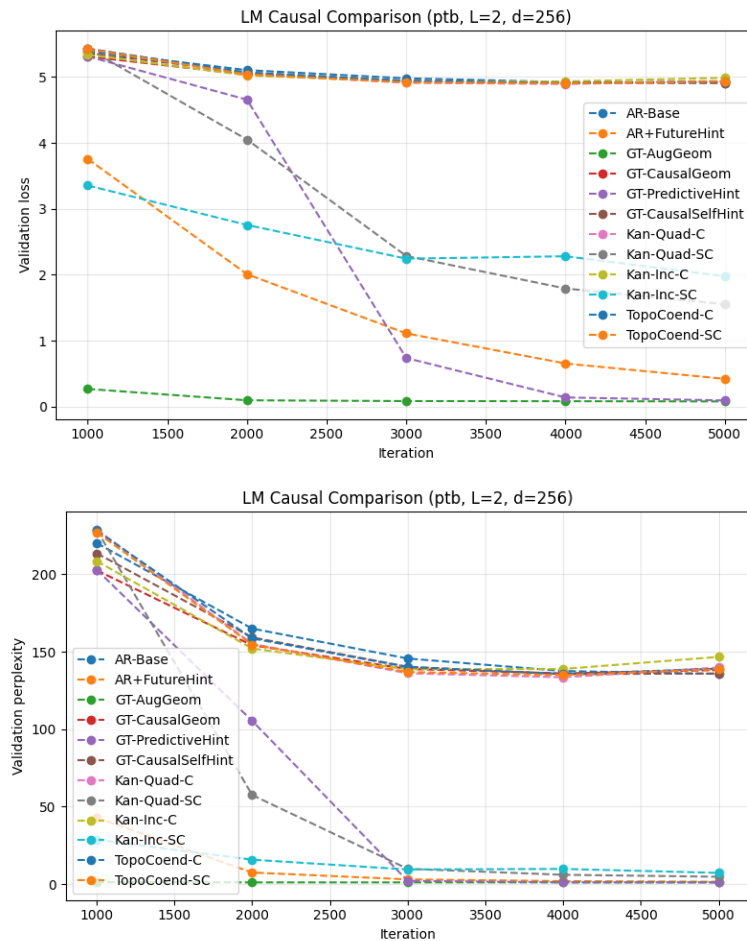


Figure 26: PTB ($L=2$, $d = 256$) validation loss and perplexity on Blackwell GPU. Strict-causal models cluster relatively closely, while future predicted- carrier A(P) variants achieve much lower teacher-forced perplexity. Both A(P) and direct gold-future variants are leakage controls.

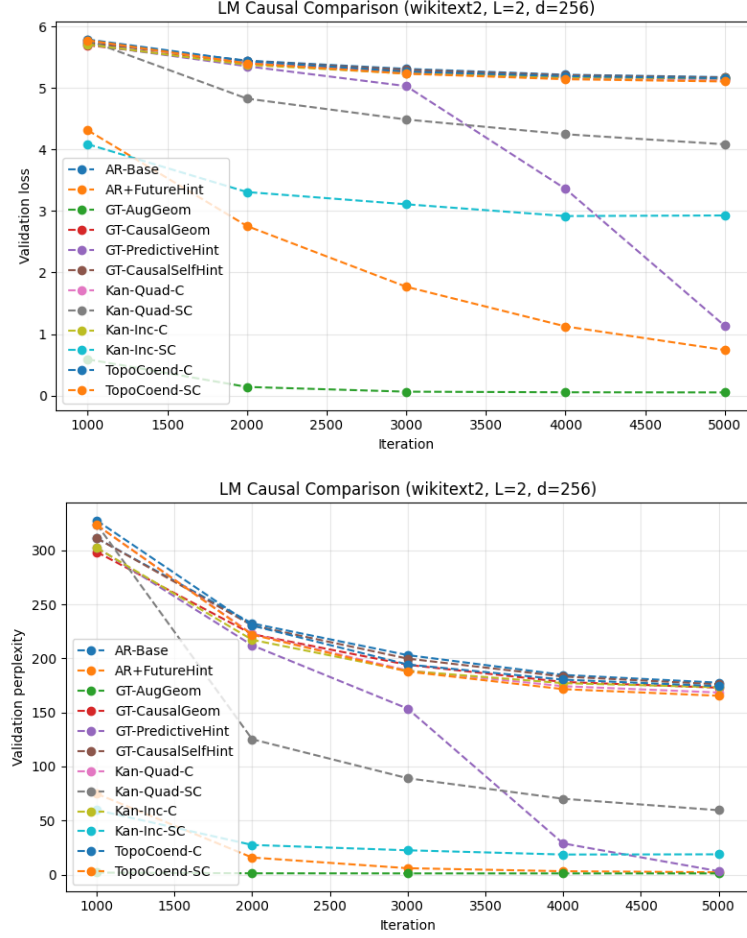
strict-causal regime. Kan-Inc-A(P) exploits this path especially strongly, while Kan-Quad-A(P) also does so. These results do not separate valid self-conditioning from leakage; implementation provenance classifies both as future-informed.

INTERPRETATION. WikiText-2 is the first dataset on which the advantage of strict-causal Quadratic KET becomes visible. This is a dataset-specific point estimate. Because corpus identity, size, and difficulty change together, it does not by itself establish a scaling law for global simplicial pooling.

WikiText-103

The figure below presents learning dynamics on WikiText-103, which provides a more demanding test of long-range structure and scaling.

STRICT-CAUSAL REGIME. On WikiText-103, Quadratic KET again



achieves the lowest point estimate among the certified strict-causal models considered. GT and incidence-restricted KET remain competitive. The reported TopoCoend- C^+ curve is numerically between these models, but its boundary defect prevents treating it as a certified causal comparison. Compared to PTB and WT2, the advantage of global simplicial pooling appears more pronounced at this scale.

FUTURE PREDICTED-CARRIER DIAGNOSTIC. The A(P) variants again obtain much smaller teacher-forced perplexities than their strict-causal counterparts. Kan-Inc-A(P) exploits the future-informed path more completely than Kan-Quad-A(P) in the recorded run, while TopoCoend-A(P) changes more modestly. This ranks leakage exploitation under these implementations; it does not rank valid language models.

INTERPRETATION. WikiText-103 supplies two observations in this run: quadratic KET has the lowest strict-causal point estimate among the com-

Figure 27: WikiText-2 (L=2, $d = 256$) validation loss and perplexity on Blackwell GPU. Quadratic KET achieves the best strict-causal perplexity in this comparison, while future predicted-carrier variants expose a strong leakage path. Direct gold-noncausal variants provide an even stronger leakage control.

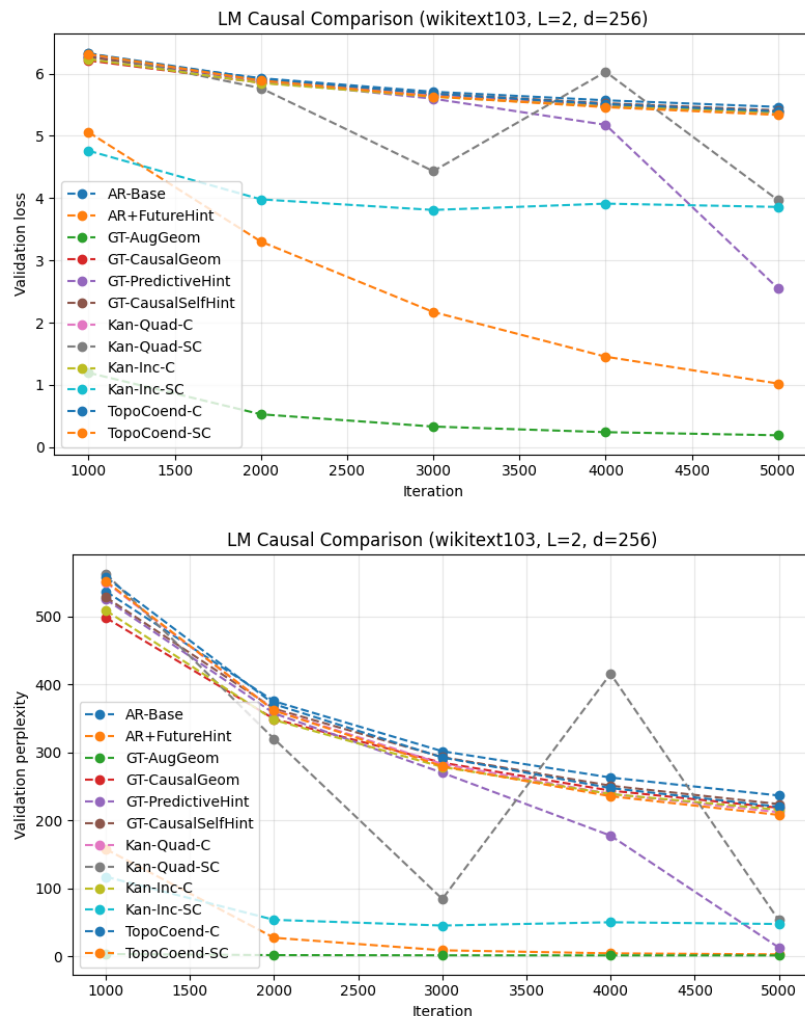


Figure 28: WikiText-103 (L=2, $d = 256$) validation loss and perplexity on Blackwell GPU. Quadratic KET achieves the lowest perplexity among the certified strict-causal models considered. Future predicted-carrier A(P) variants obtain much lower teacher-forced perplexity, while direct gold-noncausal variants approach 1. Both are leakage controls rather than autoregressive results.

pared models, and provenance violations can dominate apparent architectural effects. Cross-dataset values alone do not establish a scaling law.

Main Results

Strict-Causal Comparison

The table summarizes the certified strict-causal comparison across PTB, WikiText-2, and WikiText-103, while retaining one explicitly marked TopoCoend implementation diagnostic.

Several patterns are visible.

PTB. On the smallest corpus, the autoregressive baseline remains strongest.

Model	PTB		WikiText-2		WikiText-103	
	Test PPL	Iter/s	Test PPL	Iter/s	Test PPL	Iter/s
AR-Base	126.32	34.31	163.70	7.51	232.52	0.78
GT-CausalGeom	127.06	31.86	158.05	7.11	215.69	0.78
Kan-Quad-C	130.39	15.25	155.27	4.00	210.30	0.57
TopoCoend-C [†]	129.65	21.86	160.29	4.81	218.13	0.59
Kan-Inc-C	135.86	26.14	160.79	5.19	213.76	0.59

Quadratic KET is competitive but does not dominate, indicating that the additional expressivity of global simplicial pooling is not yet fully rewarded at this scale.

WIKITEXT-2 AND WIKITEXT-103. On both larger corpora, Quadratic KET achieves the best strict-causal perplexity among the models considered. This motivates, but does not establish, the hypothesis that global simplex pooling may be useful on these larger corpora. A scaling claim requires matched multi-seed sweeps in which model and data factors are varied independently.

INCIDENCE-RESTRICTED KET. Kan-Inc-C remains competitive while reducing computational cost relative to Quadratic KET. It is a practical linear-time alternative to full simplicial aggregation; the table does not prove approximation error bounds or functional inclusion.

TOPOCOEND BOUNDARY DIAGNOSTIC. The intended-causal TopoCoend-C[†] row improves over the autoregressive baseline on WT2 and WT103, but it cannot be counted as causal evidence. At $t = 0$, self-neighbors are excluded and no past neighbor exists. The evaluated code masks every candidate with a large finite value, then applies top- k and softmax; because all selected logits are equal, it assigns nonzero mass to gathered future values. A corrected implementation must return a zero message or preserve the residual whenever the admissible set is empty, followed by rerunning the comparison.

Future Predicted-Carrier Diagnostic

The table below reports the future predicted-carrier A(P) comparison.

The A(P) results make the provenance point especially clear.

Model	PTB Test PPL	WT2 Test PPL	WT103 Test PPL
Kan-Quad-A(P)	4.28	54.24	51.89
Kan-Inc-A(P)	6.53	18.29	47.17
TopoCoend-A(P)	128.85	152.71	206.33

Table 9: Intended strict-causal comparison ($L = 2, d = 256$, context = 128). Test perplexity is lower-is-better. Iterations per second are end-to-end throughput on the Blackwell GPU runs. Each corpus column is drawn from one internally consistent run; values are single-run point estimates and no uncertainty intervals are available. [†]The evaluated TopoCoend-C implementation leaks at the $t = 0$ empty-neighborhood boundary and is retained only as an implementation diagnostic.

Table 10: Future predicted-carrier A(P) comparison ($L = 2, d = 256$, context = 128). Each target may receive a detached carrier computed at a future teacher-forced position. These are leakage diagnostics, not autoregressive perplexities.

- A stop-gradient boundary does not remove future information from the forward computation.
- The Kan variants exploit the future-carrier path most strongly in these runs, especially incidence-restricted KET on WT2 and WT103.
- TopoCoend-A(P) exploits the same path less completely under the recorded configuration; this is not evidence about autoregressive quality.

Diagnostic Gold-Noncausal Regimes

For completeness, we also evaluate explicitly invalid regimes that mix gold future information through teacher-forced hidden states. These models collapse toward perplexity near 1, exactly as predicted by the causality analysis. They are therefore useful only as diagnostics, not as fair autoregressive baselines.

Hardware Effects

The runtime cost of Quadratic KET depends strongly on hardware. On consumer-class accelerators such as Apple MPS, full global simplicial pooling incurs a substantial slowdown relative to the baseline. On Blackwell GPUs, dense matrix operations are much more effectively accelerated, narrowing the practical gap between Quadratic KET and the lighter-weight models. This suggests that the viability of global Kan pooling depends not only on modeling benefit but also on accelerator characteristics.

Discussion of the Experimental Findings

The experiments support four high-level conclusions.

1. STRICT-CAUSAL QUADRATIC KET IS COMPETITIVE IN THESE RUNS. On PTB, strict-causal Quadratic KET is competitive but not dominant. On WikiText-2 it slightly outperforms GT, and on WikiText-103 it again achieves the best strict-causal perplexity among the models considered. This motivates a scaling experiment; it does not establish one because the three datasets are different tasks and the values lack replication.
2. INCIDENCE RESTRICTION IS AN EFFICIENT APPROXIMATION. Incidence-restricted KET captures much of the benefit of full global simplicial pooling while remaining closer in runtime to GT. This supports comparing GT with the incidence-restricted branch of a broader Kan-style design family, without proving an inclusion theorem.
3. INFORMATION REGIME DOMINATES ARCHITECTURE. The largest perplexity improvements do not come from changing the neighborhood

system alone. They come from changing the information regime from strict causality to a future-informed A(P) computation and therefore cannot be counted as language modeling gains.

4. **GOLD NONCAUSAL MIXING COLLAPSES.** Whenever noncausal structure is enriched with hidden states encoding gold future tokens, perplexity collapses toward 1. Together with the A(P) audit, this validates the need to inspect both carrier construction and target-relative source index, not only neighborhood labels.

Limitations and Future Directions

The results of this chapter are encouraging, but they also clarify the current limits of the framework and the most important directions for future work.

OPTIMIZATION AND INDUCTIVE BIAS. Quadratic KET is often slower to optimize than GT and the autoregressive baseline, especially early in training. This is consistent with its greater expressivity: the model introduces a second global kernel over simplicial objects and therefore weakens the strong local inductive bias enjoyed by incidence-restricted models. Future work should explore windowed, sparsified, or low-rank Kan kernels that preserve some of the expressive gain of global simplicial pooling while improving optimization stability and throughput.

SELF-CONDITIONING DYNAMICS. The reported future-carrier runs are not valid evidence for autoregressive self-conditioning. A valid experiment must use shifted causal carriers, sequentially generated carriers, or carriers re-computed separately from each target prefix, and must evaluate free-running generation as well as teacher-forced perplexity. Only then is it meaningful to study temperature, fixed points, schedules, or stability.

RICHER SOURCE CATEGORIES. Most experiments in this chapter focus on edge-only simplicial structures for computational efficiency. This leaves open a broader design space. Higher simplices, task-specific motifs, and domain-adapted indexing categories may prove more beneficial in settings where explicit relational structure is central. The most natural application domains include symbolic reasoning, scientific graphs, program structure, and multimodal systems with nontrivial higher-order relations.

TOPOCOEND AND LEARNED NEIGHBORHOOD STRUCTURE. TopoCoend provides a useful contrast to KET by replacing prescribed simplicial structure with learned geometric neighborhoods. In the reported experiments, the intended-causal TopoCoend curve is competitive, but the $t = 0$ empty-neighborhood defect prevents a causal interpretation; its A(P) leakage diag-

nostic changes less than the corresponding Kan diagnostics. Understanding a corrected strict-causal architecture requires repairing that boundary case and then running replicated ablations of learned coordinates, neighbor selection, and value transport.

BEYOND LEFT KAN EXTENSIONS. This chapter emphasizes left Kan extensions because the primary architectural focus is contextual aggregation. But the broader conceptual program is richer. Right Kan extensions, ends, profunctors, and bidirectional Kan constructions offer exact categorical targets for compatible completion. Connecting them to denoising or diffusion additionally requires a noise process, probabilistic semantics, and an approximation or optimization rule.

TOWARD A UNIVERSALITY THEOREM. A theoretical conjecture suggested by this chapter is that a suitably restricted class of local structured aggregation operators may admit a universal characterization. A plausible theorem schema is that natural, local, colimit-preserving contextualization operators factor through weighted left Kan extensions, with dual refinement operators factoring through weighted right Kan extensions. Establishing such a result rigorously would place KET in direct analogy with earlier universal-property formulations in causal semantics and would provide a deeper mathematical foundation for the present design comparison.

Overall, KET and TopoCoend should not be judged only by whether they uniformly dominate existing baselines. Their present contribution is a disciplined design comparison: structured aggregation can be measured against universal extension principles, while provenance auditing separates architectural effects from leakage.

Conclusion

This chapter introduced two categorical generalizations of Transformer-style language models: *Kan Extension Transformers* (KET) and *Topological Coend Transformers* (TopoCoend). The central claim of the chapter is not merely architectural. It is that attention, geometric mixing, simplicial aggregation, and learned-neighborhood pooling can all be compared as instances of a more general problem: *how should local representations be extended across structured domains in a principled way?*

KET answers this question using Kan-style weighted aggregation over simplicial incidence categories. In the special representable, functorial cases identified above, this aggregation is an enriched left Kan extension; the generic neural update is not claimed to satisfy that universal property. TopoCoend answers it using weighted coend-style aggregation over learned latent-space neighborhoods. In both cases, the key move is to shift from ad hoc mixing rules to a universal-extension perspective. Under this view, ordi-

nary attention, GT, and simplicial or learned-neighborhood models become comparable weighted-aggregation templates. Strict categorical special-case claims require explicit functors and commuting comparison maps.

A second major contribution of the chapter is the clarification of *information regimes*. We distinguished three regimes: strict-causal, gold-noncausal (invalid), and predict-and-detach self-conditioned. The chapter’s theoretical and empirical results agree sharply on this point. If noncausal structure is enriched with hidden states encoding gold future tokens, perplexity collapses through future-token leakage. If the same structural channels are enriched through detached predictive carriers, direct gold hidden-state transport and the auxiliary gradient path are removed. Future-indexed carriers nevertheless remain noncausal in the forward pass unless they are recomputed from each target prefix.

The experiments on PTB, WikiText-2, and WikiText-103 reveal a consistent pattern. On smaller corpora, strict-causal quadratic KET is competitive but does not always surpass simpler baselines. As scale increases, however, global simplicial pooling becomes more advantageous, and quadratic KET achieves the best strict-causal perplexity among the models considered in the present runs. Incidence-restricted KET provides an efficient linear-time approximation, while TopoCoend offers a complementary learned-geometry route to structured aggregation. Across all datasets, the largest numerical changes occur after changing the *information regime*: future-indexed predicted carriers give earlier targets access to teacher-forced future information. These changes diagnose leakage sensitivity rather than domination of strict-causal models.

The broader lesson is therefore twofold. Empirically, *information regime often matters more than architectural detail*. Conceptually, Kan extensions and coends provide a principled language for stating the exact target. The distinction between a certified universal construction and a Kan-style neural analogue prevents category-theoretic terminology from substituting for a theorem.

Kan Extension Transformers do not replace existing architectures. Rather, they provide a unifying conceptual bridge linking attention, geometric mixing, simplicial aggregation, and self-conditioning under a single categorical lens. That lens, we believe, will be useful not only for language modeling but for structured deep learning more broadly.

Additional Details

This appendix collects algorithmic and implementation details used in the experiments.

Algorithm 11: Quadratic Kan Extension Block

Input: token states $h_{0:S-1}$, simplex set $\mathcal{I}$, value base v , optional causal mask**Output:** updated states $h'_{0:S-1}$

```

1: Construct simplex values  $V(\sigma)$  for all  $\sigma \in \mathcal{I}$ 
2: Construct simplex keys  $K_\sigma = W_K V(\sigma)$ 
3: for  $t = 0$  to  $S - 1$  do
4:    $Q_t \leftarrow W_Q h_t$ 
5:    $\mathcal{I}_t \leftarrow \{\sigma \in \mathcal{I} : \max(\sigma) \leq t\}$  if causal mask is enabled; otherwise
      $\mathcal{I}_t \leftarrow \mathcal{I}$ 
6:    $w(t, \sigma) \leftarrow \text{softmax}_{\sigma \in \mathcal{I}_t}(Q_t^\top K_\sigma)$ 
7:    $m_t \leftarrow \sum_{\sigma \in \mathcal{I}_t} w(t, \sigma) V(\sigma)$ 
8:    $h'_t \leftarrow \text{LN}(h_t + \text{MLP}(m_t))$ 
9: end for
10: return  $h'_{0:S-1}$ 

```

Quadratic Kan Extension Block

This block implements global simplicial pooling. Its interaction cost is $O(S|\mathcal{I}|)$: it is quadratic when the edge-level simplex list has $O(S)$ objects, but can be higher when a larger collection of higher-dimensional simplices is enumerated.

Incidence-Restricted Kan Block

Algorithm 12: Incidence-Restricted Kan Block (Edge-Only)

Input: token states $h_{0:S-1}$, value base $v_{0:S-1}$, regime flag**Output:** updated states $h'_{0:S-1}$

```

1: for  $t = 1$  to  $S - 1$  do
2:    $e_t \leftarrow \psi([v_{t-1}, v_t])$ 
3: end for
4: for  $t = 0$  to  $S - 1$  do
5:    $m_t \leftarrow 0$ 
6:   if  $t \geq 1$  then
7:      $m_t \leftarrow m_t + \phi(e_t)$ 
8:   end if
9:   if regime is noncausal and  $t + 1 \leq S - 1$  then
10:     $m_t \leftarrow m_t + \phi(e_{t+1})$ 
11:   end if
12:    $h'_t \leftarrow \text{LN}(h_t + m_t)$ 
13: end for
14: return  $h'_{0:S-1}$ 

```

This block implements the linear-time approximation in which only inci-

dent edges contribute to each token update.

Predict–Detach Carrier Construction

For the reported predict–detach regimes, the value base is obtained by mapping teacher-forced hidden states through the model’s predicted token distribution and then stopping the auxiliary gradient:

$$\hat{e}_t = \text{detach}\left(\text{softmax}(\ell_t/T) E\right), \quad \ell_t = W_o h_t. \quad (17)$$

These carriers may then be used in either quadratic or incidence-restricted Kan aggregation, as well as in TopoCoend neighborhood construction. At their own index they are prefix-derived; a future-indexed carrier consumed by an earlier target nevertheless remains future-informed. A causal version must shift carriers to already generated positions or recompute each carrier from the receiving target’s prefix.

Required Leakage Audit

Shuffling targets is at most a stress test; it cannot certify causal validity. A sufficient implementation audit for a target logit must establish:

1. the backward provenance cone contains no input beyond the target prefix;
2. training and evaluation use the same shift and carrier protocol;
3. changing future teacher-forced inputs while holding the prefix fixed leaves the target logit invariant up to numerical tolerance;
4. free-running evaluation does not rely on unavailable simultaneous future carriers.

The first item is structural; the perturbation and free-running checks are empirical safeguards. Failure to collapse under a shuffle does not prove the absence of leakage, and no such shuffle result is used here to certify the reported A(P) variants.

Hyperparameters and Training Details

Unless otherwise stated, the headline language-model experiments use:

- optimizer: AdamW
- learning rate: 3×10^{-4}
- weight decay: 10^{-5}
- batch size: 32
- context length: 128

- number of layers: 2
- hidden dimension: 256
- attention heads: 4
- training iterations: 5000

Future predicted-carrier diagnostics use temperature $T = 1.0$ unless otherwise noted. Gradients are clipped to norm 1.0. The displayed comparison uses a common configured seed within each run group, but no multi-seed uncertainty is reported.

Hardware and Runtime Measurement

Experiments were conducted on:

- Apple MPS hardware
- NVIDIA DGX Spark (Blackwell GPU)

Runtime is measured end-to-end by wall-clock time. Iterations per second are computed as

$$\text{iters/sec} = \frac{\text{number of iterations}}{\text{total wall-clock seconds}}.$$

The recorded Blackwell throughput narrows the runtime gap relative to the MPS runs, but hardware conclusions remain implementation- and configuration- specific.

Summary

Kan extensions, coends, limits, and colimits are central constructions in category theory and are treated in standard texts¹⁸. What is new in this chapter is not the mathematics of Kan extensions themselves, but their use as a design language for structured Transformer architectures.

The categorical perspective developed here should be read primarily as a conceptual framework. It explains why attention, geometric mixing, simplicial aggregation, and learned-neighborhood pooling belong to the same computational family of structured extension operators. At the same time, efficient implementations that scale to larger models and corpora will require additional algorithmic ideas, including sparse kernels, approximate neighborhood search, and more structured source categories.

An ambitious theoretical direction is to make precise the possibility that a restricted KET-like family admits a universal characterization of local structured aggregation, just as earlier categorical work sought universal formulations of causal semantics. Whether that universality can be fully established remains future work, and the experiments here do not prove it. Kan-style constructions provide a concrete candidate language in which that question can be posed.

¹⁸ Saunders Mac Lane. *Categories for the Working Mathematician*. Springer-Verlag, New York, 1971. Graduate Texts in Mathematics, Vol. 5; and E. Riehl. *Category Theory in Context*. Aurora: Dover Modern Math Originals. Dover Publications, 2017. ISBN 9780486820804. URL <https://books.google.com/books?id=6B9MDgAAQBAJ>

Suggested Reading

Kan's original paper [Kan, 1958] and the treatments in Mac Lane [1971] and Riehl [2017] provide the mathematical foundation for Kan extensions. For their role in probability, see van Belle [2024]. Prompt repetition offers a useful empirical comparison for structured extension mechanisms [Leviathan et al., 2025].

Structured Language Modeling

Two Tasks, Not One Leaderboard

The previous chapter separated categorical universal constructions from neural weighted-sum analogues and, independently, separated architecture from information provenance. The same discipline is essential here. We compare two ways to train a model over a future block, but they answer different statistical questions:

$$\text{multi-offset prediction: } x_{\leq t} \mapsto (x_{t+1}, \dots, x_{t+B}), \quad (18)$$

$$\text{conditional reconstruction: } (x_{\leq t}, \tilde{y}_t, r) \mapsto (x_{t+1}, \dots, x_{t+B}), \quad (19)$$

where $y_t = (x_{t+1}, \dots, x_{t+B})$, $\tilde{y}_t$ is a corrupted copy of that target, and r records the corruption level.

The second task receives information about the answer that the first does not. Its conditional reconstruction perplexity is therefore not an improved autoregressive perplexity. It measures how well the model reconstructs a target under a specified corruption channel. The chapter’s purpose is to make this distinction explicit, document the implemented experiment, and identify what a fair structured-completion study would require.

The Product Target and Its Factorized Realization

Let Σ be the finite token vocabulary. A clean block is an element of the product set Σ^B , and the deterministic label map has the form $C_t \rightarrow \Sigma^B$. The implemented probabilistic model does *not*, however, parameterize an arbitrary joint distribution on Σ^B . From a causal hidden state h_t , an offset embedding o_j , and a shared head, it produces B categorical marginals

$$q_{\theta,j}(\cdot \mid x_{\leq t}) = \text{softmax}(g_{\theta}(h_t + o_j)), \quad j = 1, \dots, B.$$

Training minimizes the mean cross-entropy over all valid positions and off-sets. Equivalently, the likelihood uses the conditional factorization

$$q_{\theta}(y_t \mid x_{\leq t}) = \prod_{j=1}^B q_{\theta,j}(x_{t+j} \mid x_{\leq t}).$$

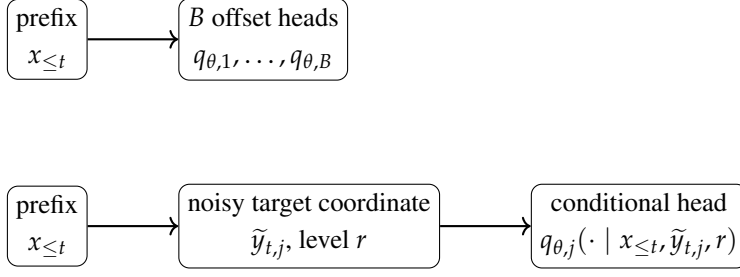


Figure 29: The two implemented tasks. Top: direct block training predicts several future offsets from the prefix. Bottom: reconstruction additionally receives a corrupted version of the target coordinate and a noise-level embedding. The latter is an easier conditional task, so its perplexity belongs on a separate scale.

The output layer therefore has $B|\Sigma|$ logits, not $|\Sigma|^B$ logits. Product notation describes the tuple-valued target; it does not prove that the neural head models dependencies among future coordinates. Testing a genuinely joint block model would require an autoregressive, energy-based, flow-based, or otherwise coupled block distribution.

The Implemented Corruption Channel

The reconstruction model uses the same prefix backbone and augments its head with an embedding of each noisy target token and an embedding of the noise step. It does not concatenate the noisy future block to the prefix and pass the resulting sequence through the backbone. At coordinate j , the head has the form

$$q_{\theta,j}(\cdot \mid h_t, \tilde{y}_{t,j}, r) = \text{softmax } g_{\theta}(h_t, o_j, e(\tilde{y}_{t,j}), \tau(r)).$$

There are eight noise steps. During training, r is sampled uniformly from $\{1, \dots, 8\}$, and each valid target coordinate is independently masked with probability

$$p_r = 0.05 + 0.45 \frac{r-1}{7}. \quad (20)$$

Unselected coordinates are left equal to their clean target. Although the source calls the replacement a mask token, the tokenizer vocabulary is not augmented: the final existing vocabulary identifier is used as the surrogate mask. In the GPT-2 tokenizer used by the loader, this is the end-of-text identifier. No random-token replacement is used in these runs.

Validation and test evaluation fix $r = 4$, hence

$$p_4 = \frac{17}{70} \approx 0.243, \quad 1 - p_4 = \frac{53}{70} \approx 0.757.$$

About three quarters of target coordinates are therefore supplied unchanged. Moreover, the reported loss averages over *all* valid coordinates, not only the masked ones. A low reconstruction perplexity can consequently be driven substantially by copying visible targets. This is legitimate for the declared conditional reconstruction task, but it is not a next-token language model score.

Algorithm 13: Factorized Multi-Offset Training

Input: token window $x_{1:T}$, offsets $1:B$, causal backbone

- 1: Construct $y_{t,j} = x_{t+j}$ wherever $t + j \leq T$
 - 2: Compute causal states $h_{1:T} = \text{Backbone}_\theta(x_{1:T})$
 - 3: **for** each valid pair (t, j) **do**
 - 4: $L_{t,j} \leftarrow g_\theta(h_t + o_j)$
 - 5: **end for**
 - 6: Minimize mean cross-entropy over all valid (t, j)
-

Categorical Interpretation: What Is Established

The clean and corrupted blocks can be organized in a category of information states, with corruption maps describing loss of information and reconstruction maps proposing sections or compatible completions. That is a useful design direction, but the implementation registers only tensors and a stochastic masking operation. It does not specify such a category, prove naturality with respect to corruption maps, or solve a universal extension problem.

Similarly, the phrase *horn filling* is only an analogy here. A simplicial horn consists of specified faces and their compatibility maps; a masked coordinate vector is not automatically a horn. To obtain a literal horn-filling experiment one would need to:

1. define the simplicial object representing a linguistic block;
2. identify which face maps are observed under each corruption;
3. verify agreement on overlaps; and
4. test existence, and where appropriate uniqueness, of fillers.

A right Kan extension is likewise an exact limit-based universal construction, not a synonym for a learned denoiser. A right-Kan account would require an indexing functor, a diagram of partial observations, and proof that the proposed completion represents the compatible-cone functor. The present neural model is best described as *completion-motivated* until those data and laws are supplied.

Algorithms Realized by the Code

The experiment crosses two backbones—Transformer and incidence KET—with the two objectives above. For every window, predictions are made at every valid position, not only at its final token.

The KET variants replace only the backbone. As Chapter 10 emphasized, the incidence update is Kan-style unless its full categorical data and universal property are verified.

Algorithm 14: Conditional Target Reconstruction

Input: token window $x_{1:T}$, offsets $1:B$, eight-step corruption schedule

- 1: Construct targets $y_{t,j} = x_{t+j}$ for all valid (t, j)
- 2: Sample one noise level $r \in \{1, \dots, 8\}$ per batch item
- 3: Independently replace each $y_{t,j}$ by the surrogate mask with probability p_r
- 4: Compute causal prefix states $h_{1:T} = \text{Backbone}_\theta(x_{1:T})$
- 5: **for** each valid pair (t, j) **do**
- 6: $L_{t,j} \leftarrow g_\theta(h_t, o_j, e(\tilde{y}_{t,j}), \tau(r))$
- 7: **end for**
- 8: Minimize mean cross-entropy over masked and unchanged coordinates

Dataset	Model	Test first-coordinate PPL	Test block PPL
PTB	TF-Block-4	144.14	316.65
PTB	KET-Block-4	143.14	315.25
PTB	TF-Denoise-4	4.17	4.54
PTB	KET-Denoise-4	4.18	4.55
WikiText-2	TF-Block-4	90.24	205.88
WikiText-2	KET-Block-4	–	–
WikiText-2	TF-Denoise-4	3.40	3.99
WikiText-2	KET-Denoise-4	–	–
WikiText-103	TF-Block-4	545.85	852.56
WikiText-103	KET-Block-4	545.10	853.32
WikiText-103	TF-Denoise-4	5.94	6.22
WikiText-103	KET-Denoise-4	5.95	6.22

Table 11: Block-4 results. “Direct PPL” and “reconstruction PPL” are different conditional metrics and must not be compared as one leaderboard. PTB values are five-seed means; the remaining admitted slices do not supply a matched multi-seed KET comparison. A dash denotes a missing admitted row.

Experimental Snapshot

The table below reports the admitted result tables currently linked to the implementation. They use local corpora tokenized by the GPT-2 tokenizer. The PTB slice is a five-seed depth-16 aggregate; the admitted WikiText slices come from different configurations, and the WikiText-2 table lacks KET rows. The table is therefore provenance evidence, not a uniform cross-dataset scaling study.

TASK EFFECT. Conditional reconstruction has far lower numerical perplexity than direct prediction. Because the reconstruction head sees a corrupted target and the metric includes unchanged coordinates, this is expected and does not establish that reconstruction is a better generative language model.

ARCHITECTURE EFFECT. The admitted snapshot does not establish a KET advantage. On the five-seed PTB aggregate, the roughly one-point block-model difference is smaller than the reported standard deviations (about 6.8

and 6.1). The denoising variants and WikiText-103 rows are effectively tied, while WikiText-2 has no admitted KET rows. Replicated matched runs are required.

NO SCALING CONCLUSION. The dataset slices differ in corpus, configuration, replication, and admitted coverage, so their PPL differences establish neither a scaling law nor the intrinsic fragility of product targets.

A Fair Completion Benchmark

The current results are a useful protocol diagnostic. A stronger experiment should separate four quantities:

1. **visible-coordinate copy accuracy**, evaluated only where the target was left unchanged;
2. **masked-coordinate reconstruction PPL**, evaluated only where the target was hidden;
3. **fully masked first-offset PPL**, which removes direct access to x_{t+1} and can be compared with a matched next-token predictor; and
4. **free-running generation**, in which no clean future target is available at inference time.

Sweep the corruption rate from zero to one using an explicit mask, matched hyperparameters, and multiple seeds. Structural claims additionally require registered relations among coordinates and direct tests of their preservation.

Conclusion

Multi-offset prediction and conditional reconstruction illuminate different interfaces: giving a model a partially uncorrupted answer changes the problem, not merely the architecture. A categorical completion theory must specify partial-information objects, refinement maps, overlap compatibility, and a universal property. The present neural mechanism motivates that theory and a fairer next experiment; it does not yet supply either result.

Suggested Reading

See Vaswani et al. [2017] for Transformers, Bradley et al. [2021] for structured semantics, Khandelwal et al. [2020] for retrieval, and May [1992] for the exact simplicial definitions behind horn filling.

Manifold Learning with Geometric Transformers

From Point Clouds to Witnessed Relations

Classical manifold learning starts with points and a notion of local neighborhood. Relational learning may start with more: typed objects, typed arrows, and witnessed higher-order relations. The important contrast is not that one family is *metric* and the other *categorical*. Rather, it is a difference in the information supplied to the learner.

A graph-based method sees the chosen vertices, edges, and edge weights.
A higher-order method may additionally see which edge configurations are declared to form triangles, joins, or commuting diagrams.

Those declarations can carry information absent from a fixed 1-skeleton. They can also be encoded in an ordinary graph by adding event nodes, relation types, or engineered features. The question is therefore empirical and protocol-dependent: under a fixed representation budget, does exposing the higher-order witnesses improve a specified task?

This chapter develops the mathematics needed to pose that question and then audits two experiments.

Experiment: higher-order relational recovery

A controlled synthetic world gives a positive result for triangle-aware features under a severe two-dimensional bottleneck. A registered MovieLens experiment gives a negative result: the same advantage does not automatically transfer to recommendation. The paired studies test task-relative utility, not a universal advantage for higher-order features.

Classical Manifold Learning Is Not One Operator

ISOMAP, locally linear embedding, Laplacian Eigenmaps, and UMAP all construct low-dimensional representations, but they do not implement one common Laplacian algorithm. Their shared practical starting point is a neighborhood graph.

ISOMAP estimates geodesic distances by shortest paths in a neighborhood graph and applies multidimensional scaling.

LLE learns local reconstruction weights and seeks coordinates that preserve those weights.

Laplacian Eigenmaps uses low-eigenvalue eigenvectors of a graph Laplacian [Belkin and Niyogi, 2003].

UMAP constructs local fuzzy memberships, combines them into a fuzzy weighted graph, and optimizes a low-dimensional fuzzy graph [McInnes et al., 2018, Spivak, 2009].

Thus UMAP has a simplicial-set motivation, but its standard computational objective compares weighted 1-skeleta. It is not simply spectral decomposition of L_0 , and the phrase *0-form geometry* describes its pairwise computational interface rather than its full mathematical motivation. Likewise, supplying triangle-derived features to UMAP can make higher-order information available to a graph-based optimizer.

Boundary Operators and Hodge Laplacians

Let K be a finite oriented simplicial complex. Write $C_k(K; \mathbb{R})$ for its k -chains and

$$B_k : C_k(K; \mathbb{R}) \longrightarrow C_{k-1}(K; \mathbb{R})$$

for the matrix of the boundary map. The chain identity is

$$B_k B_{k+1} = 0.$$

For an unweighted complex with the standard inner products, the Hodge Laplacian on k -cochains is

$$L_k = B_k^\top B_k + B_{k+1} B_{k+1}^\top.$$

At vertices,

$$L_0 = B_1 B_1^\top,$$

which is the ordinary unweighted graph Laplacian. At edges,

$$L_1 = B_1^\top B_1 + B_2 B_2^\top$$

combines lower adjacency through vertices with upper adjacency through triangles. Weighted complexes require corresponding weighted adjoints; the displayed formulas should not be reused unchanged after arbitrary weights are introduced.

The distinction in degree matters. L_0 acts on vertex signals, whereas L_1 acts on edge signals. An L_1 eigenspace is therefore not automatically a node embedding comparable with UMAP. Such a comparison needs an explicit readout or pooling map from edge coordinates back to the entities being evaluated. The registered comparator below uses one particular pooled readout; its failure does not establish that all Hodge feature maps fail.

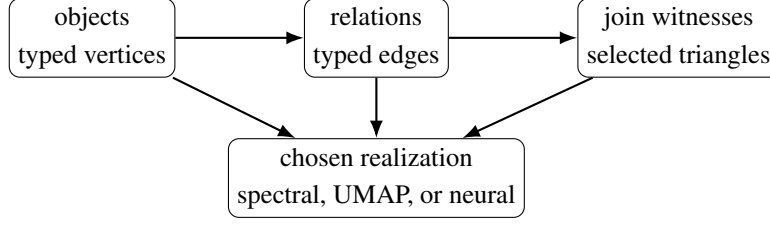


Figure 30: Higher-order relational learning has two separate design decisions: which witnesses are declared, and how those declarations are realized numerically. Category theory helps type the first decision; it does not choose the second or guarantee improved prediction.

From a Typed Instance to a Simplicial Complex

A finite sketch can declare object types and arrows such as

$$A \xrightarrow{R} B, \quad B \xrightarrow{S} C, \quad A \xrightarrow{T} C.$$

An instance assigns sets to the object types and functions or relations to the arrows. A witnessed triple (a, b, c) may be registered whenever

$$R(a, b), \quad S(b, c), \quad T(a, c)$$

all hold. After choosing orientations, these witnesses define 2-simplices and hence incidence matrices B_1 and B_2 .

This construction must be stated rather than assumed. Three pairwise edges do not intrinsically determine a semantic triangle, and a semantic triangle is not automatically a proof that $T = S \circ R$. The data designer must say which triples are witnesses, which faces they have, and whether multiple witnesses with the same boundary remain distinct. A simplicial complex collapses duplicate simplices; a simplicial set or typed event construction can retain them.

What the Geometric Transformer Realizes

A triangle-aware message-passing layer may update a vertex state by

$$h_v^{(\ell+1)} = h_v^{(\ell)} + \sum_{e \ni v} \phi_\ell(h_e^{(\ell)}, \text{type}(e)) + \sum_{\tau \ni v} \psi_\ell(h_\tau^{(\ell)}, \text{type}(\tau)).$$

An edge-only ablation removes the final sum. This creates a controlled architectural comparison when both variants receive the same vertices and typed edges and only one receives triangle witnesses.

The name *Geometric Transformer* does not itself imply that the model computes a Hodge Laplacian, preserves simplicial identities, or realizes a categorical universal property. Those claims require additional equations. Similarly, Diagrammatic Backpropagation adds structural content only when explicit path residuals such as

$$r_\Delta(a) = T(a) - (S \circ R)(a)$$

are registered and measured. Generic triangle messages and explicit commutativity penalties are related design choices, not synonyms.

Representation	Accuracy	ROC AUC
PCA, flattened relations	0.485 ± 0.029	0.512 ± 0.035
UMAP, R -Jaccard	0.506 ± 0.029	0.490 ± 0.040
UMAP, typed edges	0.542 ± 0.048	0.553 ± 0.064
UMAP, triangle features (2D)	0.545 ± 0.060	0.568 ± 0.101
Normalized L_0	0.517 ± 0.014	0.517 ± 0.048
Pooled Hodge L_1	0.519 ± 0.011	0.499 ± 0.051
DB+GT, edge-only (64D)	0.508 ± 0.027	0.509 ± 0.027
DB+GT, triangle-aware (64D)	0.892 ± 0.020	0.901 ± 0.022
DB+GT, triangle-aware, PCA to 2D	0.906 ± 0.014	0.905 ± 0.015

Table 12: Controlled relational recovery at the registered $p_{\text{close},1} = 0.15$ endpoint. All methods use held-out five-fold probes; values are mean $\pm$ sample standard deviation across data seeds $0, \dots, 9$. The Hodge comparator was specified after the primary endpoint but before its outcomes were observed.

Controlled Relational Recovery

World and preregistered endpoint

The controlled world contains 400 objects of type A , 120 of type B , and 120 of type C , divided into four groups. Relations R and S have fixed degree eight with five percent cross-group noise. Each A -object also has a hidden binary mode. In mode 0, relation T closes through $S \circ R$ with probability 0.95; at the registered primary endpoint, mode 1 closes with probability 0.15. The target is to recover the mode of held-out A -objects.

Representations are constructed without mode labels. Evaluation uses a stratified five-fold logistic probe. Reported values are means and sample standard deviations across ten paired data seeds. The DB+GT realization uses four untrained, randomly initialized message-passing layers with 64 coordinates and a fixed model seed; only the held-out probe is supervised. It is therefore a random-feature relational encoder, not a trained state-of-the-art classifier.

The two-dimensional triangle-aware DB+GT representation exceeds R -only UMAP by 0.400 accuracy, with paired bootstrap 95 percent interval $[0.380, 0.421]$. It exceeds two-dimensional triangle-feature UMAP by 0.361, with interval $[0.321, 0.390]$. These comparisons establish a narrow result: this realization exposes the registered closure signal efficiently under the chosen low-dimensional readout.

The decisive capacity diagnostic

A post-confirmatory dimension sweep changes the interpretation. When UMAP receives the triangle features directly, its accuracy rises from 0.545 ± 0.060 in two dimensions to 0.839 ± 0.074 in six dimensions and 0.906 ± 0.014 in eight dimensions. Thus the experiment does not show that graphs or UMAP cannot represent join information. It shows that the tested DB+GT feature map packages that information more effectively under the two-dimensional bottleneck.

Method	NDCG@10	Recall@10	RMSE
Popularity	0.069 ± 0.000	0.082 ± 0.000	1.232 ± 0.000
Global/user/item bias	0.034 ± 0.000	0.041 ± 0.000	1.017 ± 0.000
Matrix factorization, 2D	0.030 ± 0.004	0.032 ± 0.002	1.000 ± 0.005
Triangle-feature UMAP, 2D	0.008 ± 0.011	0.009 ± 0.012	1.136 ± 0.012
DB+GT, edge-only, 2D	0.025 ± 0.002	0.029 ± 0.003	1.159 ± 0.003
DB+GT, triangle-aware, 2D	0.025 ± 0.011	0.026 ± 0.013	1.156 ± 0.001

Table 13: Selected frozen MovieLens test results, mean $\pm$ sample standard deviation across five representation seeds. The experiment did not pass its preregistered real-data gate.

The pooled Hodge baseline remains near chance. This is evidence about the chosen edge spectral features and pooling rule, not about Hodge theory in general. A learned simplicial network, a different kernel, or a different edge-to-vertex readout could behave differently.

MovieLens: A Registered Negative Result

The real-data study uses MovieLens 100K: 100,000 ratings from 943 users on 1,682 movies. Events are split chronologically within each user: the first 80 percent define all representations, the next 10 percent select probe regularization, and the final 10 percent are held out for testing.

The typed schema contains `User`, `Movie`, and `RatingEvent`, with maps from an event to its user, movie, and value. Each training event supplies a witnessed user–movie–event triangle. All manifold comparisons expose two user coordinates and two movie coordinates to a shared rating probe. The primary endpoint is NDCG@10; RMSE is a secondary gate.

Triangle-aware DB+GT does not improve on its edge-only ablation, and both trail the popularity baseline on ranking. More importantly, the registered RMSE criterion fails. The earlier single-run plots and approximate claims of 1.10 versus 1.14 RMSE are therefore not retained as evidence of a generalization advantage.

This negative result is scientifically useful. Higher-order witnesses can make a controlled latent relation recoverable without yielding a useful recommendation representation. Success depends on the target, the construction of witnesses, the representation dimension, the readout, and the amount of data. *Relational* is not itself an inductive-bias guarantee.

Categorical and Coalgebraic Claim Boundaries

A categorical formulation can organize the schema and the comparison maps. For example, if a sequential system is a coalgebra

$$\gamma : X \longrightarrow TX,$$

and F maps states into a representation category, compatibility with a represented dynamics $\hat{\gamma} : FX \rightarrow \hat{T}(FX)$ requires an explicit comparison

between FT and $\widehat{TF}$ and a commuting naturality square. Merely training an embedding on transition triples does not establish such a lifting.

The same caution applies to the word *manifold*. A learned point cloud is not proved to be a smooth manifold, and a simplicial complex is not automatically a triangulation of one. In this chapter, *relational manifold learning* names a representation-learning program: construct geometry from typed relational witnesses and audit which structures and tasks it preserves.

A Reproducible Comparison Contract

A persuasive higher-order comparison should register:

1. the typed schema and the rule that creates each higher simplex;
2. the information supplied to every baseline, including engineered triangle features and event-node graph encodings;
3. the degree on which each operator acts and every pooling or readout map;
4. matched representation dimension, supervision, optimization budget, and parameter count where applicable;
5. held-out downstream metrics with repeated seeds and uncertainty;
6. structural audits such as $B_k B_{k+1} = 0$, incidence residuals, or declared path-equation residuals; and
7. negative controls in which triangle witnesses are shuffled, omitted, or made irrelevant to the target.

Information provenance from Chapter 9 remains orthogonal to this contract. Future-informed features do not become causal merely because they are attached to a simplex. Likewise, KET-style aggregation from Chapter 10 does not become a Kan extension merely because it is applied to a relational neighborhood.

Conclusion

Higher-order witnesses enlarge the information interface available to a learner. Boundary operators, Hodge Laplacians, and simplicial message passing offer distinct ways to process that interface; none should be collapsed into the others.

The controlled result shows that a triangle-aware random-feature encoder can recover a closure mode that tested edge-only representations miss under a two-dimensional bottleneck. The dimension diagnostic shows that a feature-engineered UMAP can recover the same information when given enough coordinates. The MovieLens result shows that neither fact guarantees improved recommendation. This is a more durable conclusion than a universal

method hierarchy: typed higher-order structure is a testable inductive bias whose value must be demonstrated for each task.

Suggested Reading

For Laplacian Eigenmaps and manifold regularization, see [Belkin and Niyogi \[2003\]](#), [Belkin et al. \[2006\]](#); for out-of-sample comparisons of classical methods, see [Bengio et al. \[2004\]](#). UMAP is developed by [McInnes et al. \[2018\]](#), with fuzzy simplicial background in [Spivak \[2009\]](#). For simplicial topology and homology, see [Ghrist \[2014\]](#) and [Edelsbrunner et al. \[2002\]](#).

Mean-Field Theory of Geometric Transformers

This chapter asks a deliberately narrower question than the preceding architectural discussion: what can a local, wide-network calculation actually say about the order sensitivity of two residual updates? The answer is useful but bounded. At small residual step size, the leading obstruction is the Lie bracket of the two update vector fields. In a solvable random-linear model its expected squared norm remains nonzero at infinite width. Layer normalization and geometric transport change this quantity through explicit Jacobian and spectral factors, but neither mechanism guarantees a universal reduction.

This is therefore a *mean-field program* for Geometric Transformers, not a closed-form infinite-width theory of attention during training. The exact statements below identify what is proved by the local model, what is merely an architectural hypothesis, and what the available experiments can and cannot support.

The Object Being Measured

Let $T_1, T_2 : X \rightarrow X$ be two maps that can genuinely be evaluated in either order on the same state space. For $x \in X \subseteq \mathbb{R}^d$, define

$$r_{12}(x) = T_1(T_2(x)) - x, \quad r_{21}(x) = T_2(T_1(x)) - x.$$

The diagnostic used in the Geometric Transformer experiments is

$$C_\Delta(T_1, T_2; x) = \frac{\|r_{12}(x) - r_{21}(x)\|^2}{\|r_{12}(x)\|^2 + \|r_{21}(x)\|^2 + \varepsilon}. \quad (21)$$

It is a forward-pass measure of *order sensitivity at x* . It does not contain a gradient, and it does not by itself measure how two parameter updates interfere during optimization.

When $\varepsilon = 0$ and the denominator is nonzero, the ratio lies in $[0, 2]$ by the parallelogram inequality and is invariant under a common rescaling of both residuals. With fixed $\varepsilon > 0$, that scale invariance is only approximate when the residual energy dominates ε . Furthermore, a small value is not automatically desirable: constant maps, vanishing residual updates, and excessive smoothing can all make the ratio small while destroying task performance.

The phrase *commutator energy* is retained as convenient terminology. It should not be confused with the commutator of parameter gradients, with

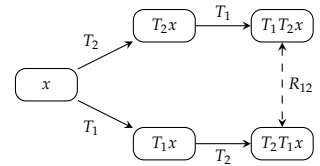


Figure 31: The measured defect compares two executable orders. A categorical interpretation requires the two paths to have been declared comparable; the numerical discrepancy does not supply that declaration.

a Čech cohomology class, or with a proof that modules ought to commute globally. Those are stronger objects requiring additional structure.

Residual Updates and the First Obstruction

Write two residual maps with explicit step sizes as

$$T_f(x) = x + \alpha f(x), \quad T_g(x) = x + \beta g(x),$$

where $f, g : \mathbb{R}^d \rightarrow \mathbb{R}^d$ are C^2 on a neighborhood of the states of interest.

Direct substitution gives the exact identity

$$\begin{aligned} T_f(T_g(x)) - T_g(T_f(x)) &= \alpha\{f(x + \beta g(x)) - f(x)\} \\ &\quad - \beta\{g(x + \alpha f(x)) - g(x)\}. \end{aligned} \quad (22)$$

Taylor expansion therefore yields

$$T_f T_g(x) - T_g T_f(x) = \alpha\beta(Df(x)g(x) - Dg(x)f(x)) + O(\alpha\beta^2 + \alpha^2\beta). \quad (23)$$

With the convention $[f, g] = Dg f - Df g$, the leading term is $-\alpha\beta[f, g](x)$. Thus the first local obstruction is a Lie bracket of *vector fields*. It is not generally the matrix commutator $Df Dg - Dg Df$.

The matrix commutator appears in the special homogeneous linear case $f(x) = Ax$ and $g(x) = Bx$:

$$T_f T_g(x) - T_g T_f(x) = \alpha\beta(AB - BA)x.$$

For affine or nonlinear maps, offsets and derivatives of the vector fields matter. This distinction is essential when attention, normalization, and feed-forward modules are replaced by local linearizations.

Equation (23) also exposes an important confound. If $\alpha = \beta = h$, the numerator of Equation (21) is $O(h^4)$ while its denominator is ordinarily $O(h^2)$. The normalized ratio is therefore $O(h^2)$ for small updates. A model can lower the diagnostic simply by shrinking its residual steps. Any empirical use must report update norms and task performance beside the ratio.

A Solvable Wide Random Model

The simplest honest mean-field calculation replaces the update fields by independent random linear maps. Let

$$f(x) = Ax, \quad g(x) = Bx,$$

where the entries of A and B are independent centered Gaussian variables with variances σ_A^2/d and σ_B^2/d , respectively. For a fixed $x \in \mathbb{R}^d$,

$$\mathbb{E} \|(AB - BA)x\|^2 = 2\sigma_A^2\sigma_B^2 \left(1 - \frac{1}{d^2}\right) \|x\|^2. \quad (24)$$

The calculation follows by expanding the two squared terms and their cross term. Independence gives $\mathbb{E}\|ABx\|^2 = \mathbb{E}\|BAx\|^2 = \sigma_A^2 \sigma_B^2 \|x\|^2$, while the shared diagonal Wick contraction is $\sigma_A^2 \sigma_B^2 \|x\|^2 / d^2$.

If $\|x\|^2/d \rightarrow q$, the expected commutator energy per coordinate converges to $2\sigma_A^2 \sigma_B^2 q$. Generic independently initialized linear update fields therefore do not become commuting merely because width tends to infinity.

This proposition is narrower than a Transformer mean-field limit. Real attention blocks contain softmax normalization, token correlations, shared weights, residual paths, and layer normalization; their fields are neither independent nor linear during training. Tensor programs and related methods provide machinery for deriving such limits, but a complete program would need to specify the joint recursion for the two composed orders rather than invoke independent random matrices after the fact ¹⁹.

What Layer Normalization Changes

For clarity, omit learned affine parameters and let

$$P = I - \frac{1}{d} \mathbf{1}\mathbf{1}^\top, \quad z = Px, \quad s(x) = \sqrt{\|z\|^2/d + \epsilon_{\text{LN}}}.$$

Then $\text{LN}(x) = z/s(x)$ and

$$D \text{LN}(x) = \frac{1}{s(x)} P - \frac{1}{d s(x)^3} z z^\top. \quad (25)$$

The centering direction is removed, the radial centered direction is strongly attenuated when ϵ_{LN} is small, and tangent directions are scaled by $1/s(x)$. Layer normalization is therefore not a scalar Jacobian shrinkage in all directions.

For an update field f , the two placements have different derivatives:

$$D\{x + \alpha f(\text{LN } x)\} = I + \alpha Df(\text{LN } x) D \text{LN}(x), \quad (26)$$

$$D\{\text{LN}(x + \alpha f(x))\} = D \text{LN}(x + \alpha f(x)) \{I + \alpha Df(x)\}. \quad (27)$$

These identities explain why normalization placement can change signal and commutator statistics. They *do not* prove a universal inequality saying that Pre-LN has lower commutator energy than Post-LN. Such an ordering depends on the input distribution, the update fields, residual scales, affine LN parameters, and the states reached during training. Mean-field analyses of Transformer normalization establish useful initialization and gradient-flow results under explicit assumptions; those results should not be silently promoted into a theorem about the diagnostic in Equation (21) ²⁰.

¹⁹ Greg Yang. Scaling limits of wide neural networks with weight sharing. *Advances in Neural Information Processing Systems*, 32, 2019. URL <https://arxiv.org/abs/1902.04760>; and Greg Yang. Tensor programs ii: Neural tangent kernel for any architecture. *Advances in Neural Information Processing Systems*, 34, 2021. URL <https://arxiv.org/abs/2006.14548>

²⁰ Ruibin Xiong, Yaru Yang, Di He, Kai Zhang, Shuxin Zheng, Hao Zheng, Chen Xing, Liangchen Liu, and Liwei Wang. On layer normalization in the transformer architecture. *arXiv preprint arXiv:2002.04745*, 2020

Transport and Spectral Smoothing

The most defensible spectral model of a geometric transport layer is an explicit Laplacian step. Let $L = L^\top \succeq 0$ be a graph Laplacian and

$$G_\tau = I - \tau L.$$

If $0 \leq \tau \leq 2/\lambda_{\max}(L)$, then G_τ is nonexpansive in the Euclidean norm. In an eigenbasis of L , it multiplies the component of frequency λ_k by $1 - \tau\lambda_k$. For the stricter range $0 \leq \tau \leq 1/\lambda_{\max}(L)$, high-frequency components are attenuated without a sign reversal, and the Dirichlet energy obeys

$$(G_\tau x)^\top L(G_\tau x) \leq x^\top Lx. \quad (28)$$

This gives an exact low-pass interpretation for G_τ . A learned GT message map has that interpretation only if its implemented operator is shown to be, or is locally approximated by, such a stable Laplacian step. Generic message passing need not be self-adjoint, positive semidefinite, conservative, or contractive.

Nor does smoothing automatically reduce order sensitivity. For a linear map A ,

$$[A, G_\tau] = -\tau[A, L],$$

which may vanish, increase, or decrease depending on whether A respects the Laplacian eigenspaces. For a K -Lipschitz nonlinear map F ,

$$\|F(G_\tau x) - G_\tau F(x)\| \leq \tau\{K\|Lx\| + \|LF(x)\|\}. \quad (29)$$

The bound proves that an infinitesimal transport step has an $O(\tau)$ defect; it does not prove that adding transport improves a pre-existing architecture. The stronger hypothesis needed by GT is *alignment*: the learned task map should approximately preserve the geometry encoded by L .

Sheaf Energy Is a Different Obstruction

A cellular sheaf on a token graph assigns vector spaces $\mathcal{F}(v)$ and $\mathcal{F}(e)$ to vertices and edges and restriction maps from incident vertices to an edge stalk. Its coboundary is

$$(\delta x)_e = \rho_{v \rightarrow e} x_v - \rho_{u \rightarrow e} x_u, \quad e = (u, v),$$

and its 0-Laplacian and normalized Dirichlet energy are

$$\Delta_{\mathcal{F}}^{(0)} = \delta^* \delta, \quad \mathcal{E}_{\mathcal{F}}(x) = \frac{\|\delta x\|^2}{\|x\|^2 + \varepsilon}. \quad (30)$$

The nullspace consists of globally compatible sections under the declared restrictions. For identity restrictions this reduces to ordinary graph smoothness

channel by channel; orthogonal restrictions yield a connection-Laplacian-type geometry.

This is a genuine linear gluing defect. The forward commutator diagnostic is a different object: it compares the order of two generally nonlinear patch maps. The two may correlate in a model where edge incompatibilities drive patch-order effects, but neither equality nor monotonicity follows from their definitions. Moreover, learned restriction maps require a nondegeneracy condition: allowing all restrictions to collapse to zero makes the sheaf energy vanish trivially. Thus a useful sheaf diagnostic needs fixed or admitted transports, a scale or isometry constraint, and a task-level check.

A Data-Limited Pilot

The available evidence consists of single-seed prefix-to-suffix experiments on WikiText-103 and WikiText-2. Each run used 1,000 training sentences, 100 validation sentences, sequences clipped to 64 tokens, four encoder/decoder blocks, width 96, four heads, and 1,000 optimization steps. The surviving artifacts contain plots and final summary values but not a complete executable snapshot with raw per-example outputs. The results therefore constitute a *data-limited pilot*, not confirmatory evidence or a reproducible sample-efficiency benchmark.

Dataset	Model	Val. CE	Token acc.	C_Δ	Dirichlet E
WikiText-103	TF Pre-LN	8.976	0.035	0.0449	1.009
	GT-Lite Pre-LN	6.557	0.181	0.0268	0.875
	GT-Full Pre-LN	6.536	0.181	0.0284	0.873
	GT-MoE Pre-LN	6.538	0.182	0.0554	0.924
WikiText-2	TF Pre-LN	7.903	0.041	0.0397	1.252
	GT-Lite Pre-LN	4.766	0.245	0.0297	1.023
	GT-Full Pre-LN	4.862	0.242	0.0309	1.054
	GT-MoE Pre-LN	4.903	0.242	0.0492	1.083

The table supports only a limited observation: under this particular harness, the GT variants achieved lower validation cross-entropy and higher token accuracy than the baseline. It does not establish that geometric transport caused the improvement. The GT models add mechanisms and possibly capacity; the comparison did not match parameter counts; only one seed is retained; and there is no sweep over sample size from which to estimate sample efficiency.

The instrumental columns also resist a causal story. On WikiText-103, GT-MoE has essentially the same task score as GT-Full while its commutator ratio is higher than the Transformer Pre-LN value. Persistent-homology summaries in the old figures likewise vary without a registered monotone prediction. Attention weights must first be converted into a specified filtration before their persistence is defined, and bar counts depend on that construc-

Table 14: Selected final values from the single-seed pilot. The table documents the observation that the tested GT variants learned the small prefix-to-suffix task faster under the recorded budget. It does not isolate the cause: parameter counts were not matched, no seed distribution or confidence interval is available, and the diagnostic is not monotone across variants.

tion. Those summaries are therefore exploratory architecture fingerprints, not evidence that one model has a better topology.

A Confirmatory Mean-Field Contract

A decisive study should separate four questions that the pilot combined.

1. *Does the theoretical limit exist?* Specify the architecture, initialization, token covariance, normalization convention, and joint recursion for both operator orders; compare finite width with the predicted limit.
2. *Does the diagnostic predict optimization within an architecture?* Vary seeds and residual scales while holding architecture fixed. Report update norms, gradient statistics, task loss, and C_Δ on held-out probe states. A pooled cross-architecture correlation is not enough.
3. *Does transport add value beyond capacity and smoothing?* Match parameter count, optimizer budget, and information interface; compare learned transport with identity, shuffled geometry, scalar diffusion, and an equally parameterized nongometric module.
4. *Is the declared categorical structure tested?* Register which paths are required to agree, measure their residuals independently of the training loss, and include tasks on which noncommutativity is useful or irrelevant. Otherwise “low obstruction” can reward a collapsed model.

Depth and width sweeps should be described as scaling experiments only after the per-depth optimization budget, parameterization, and uncertainty have been controlled. The chapter that follows applies this standard to the available depth-scaling evidence.

Conclusion

The local theory yields three durable conclusions. First, the leading order-sensitive defect of two small residual updates is their Lie bracket. Second, generic independent random linear updates retain nonzero bracket energy at infinite width. Third, normalization and geometric transport alter the defect through explicit Jacobian and spectral structure, but neither supplies a universal monotonic guarantee.

These conclusions motivate commutator energy as an instrument, not an order parameter whose minimization defines learning. A trustworthy GT theory must couple the instrument to task performance, noncollapse controls, matched architectures, and the actual diagrammatic obligations being tested. That more modest interpretation is also more useful: it turns a suggestive plot into a falsifiable program of analysis.

Suggested Reading

Mean-field analyses of signal propagation and critical initialization begin with [Poole et al. \[2016\]](#) and [Schoenholz et al. \[2017\]](#). Tensor programs provide a systematic language for wide-network limits across broader architecture classes [\[Yang, 2019, 2021\]](#). [Xiong et al. \[2020\]](#) analyzes the effect of normalization placement in Transformers. For the differential and cohomological background behind Lie brackets, Laplacians, and gluing obstructions, see [Bott and Tu \[1982\]](#) and [Ghrist \[2014\]](#).

Depth Sweeps for Geometric Transformers

The word *scaling* can describe two very different enterprises. A depth sweep evaluates several finite architectures. A scaling law fits a stable quantitative relation over a declared range and survives prediction outside the points used to estimate it. The Geometric Transformer experiments belong to the first category.

That distinction does not make them uninformative. Their most durable observation is that the recorded *average per-layer* order-sensitivity diagnostic tends to be smaller in deeper models. But the same tendency occurs in the Pre-LN baseline and in GT-Full, so it does not by itself identify a new GT scaling regime. This chapter audits the experiments, explains the aggregation and compute confounds, and turns the surviving observation into a properly testable scaling program.

What Would Count as a Scaling Law?

Let M be a metric such as validation cross-entropy, total compute, or an order-sensitivity diagnostic. A scaling claim should specify a protocol π and a response surface

$$m(L, W, D, C; \pi) = \mathbb{E}[M \mid L, W, D, C, \pi],$$

where L is depth, W width, D data, and C compute. A proposed law such as

$$m(L; \pi) = m_\infty + aL^{-\alpha} \tag{31}$$

is meaningful only after the other coordinates, the training rule, and the uncertainty model have been fixed. It should be compared with alternative functional forms and tested on held-out depths. Four plotted depths with one seed do not identify an asymptotic exponent.

The familiar neural-language-model scaling literature studies empirical power laws in loss as parameters, data, and compute vary, and emphasizes the coupling among these resources²¹. A proposed geometric scaling law has an additional burden: it must show that its structural observable predicts something beyond those ordinary resources.

²¹ Jared Kaplan, Sam McCandlish, Tom Henighan, Tom Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020; and Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022

Depth Changes More Than Depth

At fixed width, a repeated-block architecture ordinarily has parameter count

$$P(L) = P_0 + Lp_{\text{block}}$$

and per-token forward cost approximately

$$F(L) = F_0 + Lf_{\text{block}}.$$

Holding the number of optimizer steps fixed therefore gives deeper models more parameters and more floating-point work. Holding total FLOPs fixed gives them fewer updates. Holding parameter count fixed requires reducing width or block capacity, changing the hypothesis class again. None is the uniquely correct comparison, but the choice must be named:

- *iso-step* asks what a fixed number of optimizer updates achieves;
- *iso-token* fixes the number of training tokens processed;
- *iso-FLOP* fixes approximate computational expenditure;
- *iso-parameter* compares models with matched trainable capacity.

Residual parameterization adds another axis. If the effective update at each layer is multiplied by h_L , the preceding chapter shows that the normalized delta-commutator ratio is ordinarily $O(h_L^2)$ for small updates. A convention such as $h_L \propto L^{-1/2}$ can therefore make the average local ratio fall like $1/L$ even when no new alignment mechanism is present.

Averaging Is Not Accumulation

The evaluated implementation computes a ratio $c_\ell(x)$ for each encoder layer and records

$$\bar{C}_L(x) = \frac{1}{L} \sum_{\ell=1}^L c_\ell(x). \quad (32)$$

This is a legitimate average local diagnostic. It is not a measure of the total obstruction accumulated by a depth- L composition.

Three summaries answer different questions:

$$\bar{C}_L = \frac{1}{L} \sum_{\ell} c_\ell, \quad S_L = \sum_{\ell} c_\ell, \quad C_L^{\max} = \max_{\ell} c_\ell.$$

If every layer has $c_\ell \asymp 1/L$, then $\bar{C}_L \asymp 1/L$ while $S_L \asymp 1$. If path interactions compound, neither summary recovers the actual end-to-end order defect. That requires perturbing declared subpaths of the composed model and measuring their output discrepancy directly.

The mean of local ratios must not be confused with accumulated obstruction. The plots show only the former and therefore cannot adjudicate a claim about the latter.

Evaluation of the Experiment Family

The surviving Python sources and plots provide partial provenance. They are enough to recover the broad design, but not to reproduce a confirmatory study.

Artifact	Recoverable protocol	Limitation
WikiText-103 Pre-LN	20,000 training and 1,000 validation sentences; 5,000 steps; depths 2, 4, 8, 16; width 96; four heads; seed 0	One sequential sweep, no confidence intervals, no held-out depth, and increasing parameters and FLOPs with depth
WikiText-103 GT-Full	Same nominal data, step, width, and depth settings as the baseline	Extra unplotted depth-16 training runs precede the plotted sweep and advance the random and loader states; parameter counts are unmatched
WikiText-103 GT-Lite and GT-MoE	Current sources declare 5,000/100 sentences and 1,000 steps	Stored plots extend to 5,000 steps; the current source does not regenerate the displayed artifact
Penn Treebank	Baseline and GT-Full sources declare 1,000/100 sentences, 1,000 steps, and depths 2, 4, 8, 16	Exact generators for every displayed architecture are not preserved with raw logs or a run manifest

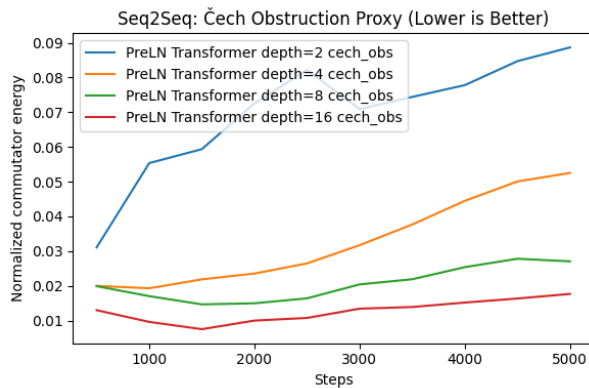
All conditions are run serially after setting one global seed. Consequently, later architectures and depths consume different portions of the pseudorandom and shuffled-data streams. This is not equivalent to paired seeds. The plots contain no uncertainty bands, and their final points are not accompanied by raw logs, model hashes, or a machine-readable manifest.

The protocol also compares different information-processing modules and parameter counts. GT-Full, GT-Lite, GT-MoE, and the baseline are not merely four parameterizations of the same function class. Cross-architecture performance gaps may reflect capacity, optimization, regularization, or implementation details in addition to geometric alignment.

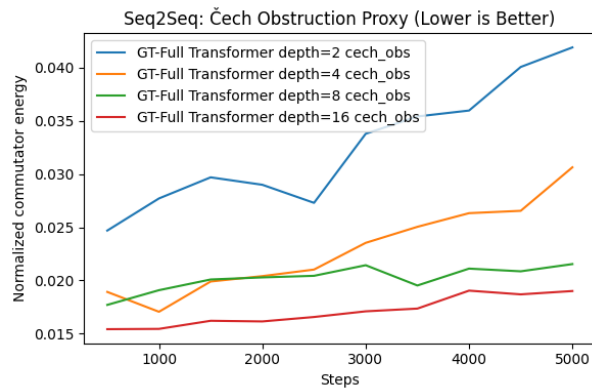
Table 15: Source-to-artifact audit of the depth sweeps. The available plotted scripts instantiate depths 2, 4, 8, 16.

What the Surviving Curves Show

The two best-traced WikiText-103 commutator plots are reproduced below. Both use the same nominal data and training horizon, although their parameter counts and random-stream positions are not matched.



(a) Pre-LN Transformer



(b) GT-Full

Figure 32: One-seed trajectories of the average per-layer delta-commutator ratio. In both architectures, deeper models generally have a smaller average local ratio, while most trajectories rise after their early minimum. The plots are descriptive evidence, not fitted scaling laws; the original “lower is better” axis wording is not an admission criterion.

Three observations survive visual and code-level inspection.

1. Within each displayed architecture, depth 16 generally has a smaller average local ratio than depth 2. This is compatible with smaller effective per-layer residual steps and is not unique to GT.
2. The shallow trajectories increase strongly with training time. The deeper trajectories also tend to increase after an early minimum, although at a smaller scale. Calling the latter “bounded” merely restates the finite vertical range of this experiment.
3. GT-Full is not uniformly lower than the baseline at every depth and training step. At the deepest final point, for example, the plotted magnitudes are similar. The figure therefore does not establish a distinct asymptotic GT regime.

The validation plots suggest that GT variants learned the particular prefix-to-suffix task more readily. Chapter 13 explains why those task results remain pilot evidence: architecture capacity was unmatched, runs were single-seed, and no sample-size curve was measured. The depth plots add no causal isolation of transport.

The persistent-homology panels are even less determinate. Their values depend on converting attention affinities into a Rips filtration, selecting tokens and layers, and averaging bar summaries. No monotone relation with validation performance was preregistered. They may be retained as exploratory fingerprints, but not interpreted as “stability by collapse” versus “stability by alignment” without independent tests of those mechanisms.

From a Depth Sweep to a Geometric Scaling Test

A confirmatory experiment can be organized as a factorial design rather than a collection of architecture-specific scripts.

1. *Freeze the task and split.* Store tokenized train, validation, and test examples together with hashes and a fixed evaluation harness.
2. *Declare the comparison budget.* Report iso-step and iso-FLOP curves separately. Include parameter count, tokens processed, wall time, and estimated FLOPs for every point.
3. *Pair randomization.* Use the same list of at least five data and model seeds for every architecture-depth cell. Reset loader and initialization generators explicitly before each run.
4. *Factor residual scale from depth.* Cross depth with at least two residual parameterizations. Log per-layer update norms so a decrease in $\bar{C}_L$ cannot be mistaken for alignment.
5. *Match controls.* Compare GT transport with identity transport, shuffled geometry, scalar diffusion, and an equally parameterized generic message module.
6. *Measure local and global defects.* Report $\bar{C}_L$, S_L , $C_L^{\max}$, and a registered end-to-end path-swap residual on held-out probe states.
7. *Fit and predict.* Fit candidate curves only on a subset such as $L \in \{2, 4, 8, 16\}$, then test predictions at new depths. Compare power law, logarithmic, saturating, and nonmonotone alternatives.

The primary GT claim should be an interaction, not a raw difference:

$$\Delta_{\text{geom}}(L) = \{M_{\text{GT}}(L) - M_{\text{control}}(L)\} - \{M_{\text{GT}}(L_0) - M_{\text{control}}(L_0)\}. \quad (33)$$

A geometric scaling advantage requires this contrast to improve predictably with depth under matched resources and to replicate at held-out depths. A single favorable GT curve is not enough.

Three Testable Hypotheses

The audited evidence motivates hypotheses rather than conclusions.

RESIDUAL-DISTRIBUTION HYPOTHESIS. At fixed architecture and task, much of the decrease in average local commutator ratio with depth is explained by smaller per-layer update norms. After conditioning on those norms, the depth coefficient should shrink.

ALIGNMENT HYPOTHESIS. If GT transport aligns task updates with a declared token geometry, it should reduce an end-to-end path-swap defect relative to equally parameterized shuffled and generic-message controls, not merely lower local smoothness energy.

PREDICTIVE-SCALING HYPOTHESIS. A genuine GT depth advantage should extrapolate to held-out depths under both iso-step and iso-FLOP budgets and should persist across seeds and at least two tasks. Failure of extrapolation would reclassify the observation as a finite architecture effect.

These hypotheses can all fail. That falsifiability is an advantage: it separates the useful structural intuition behind GT from the much stronger claim that a new universal scaling regime has already been discovered.

Conclusion

The experiments establish a finite descriptive fact: in the displayed Pre-LN and GT-Full runs, deeper networks usually have lower average per-layer order sensitivity. They do not establish an exponent, an asymptotic law, an accumulated obstruction, or a GT-specific scaling mechanism.

Recasting the chapter around that boundary improves the research program. Depth, residual scale, parameters, compute, geometry, and aggregation are now separate experimental variables. A future geometric scaling result will be strong precisely when it survives those controls and predicts unseen depths.

Suggested Reading

For empirical neural-language-model scaling laws, see [Kaplan et al. \[2020\]](#); for the coupling of parameters, data, and compute, see [Hoffmann et al. \[2022\]](#). Mean-field signal propagation and wide-network limits are developed by [Poole et al. \[2016\]](#), [Schoenholz et al. \[2017\]](#), and [Yang \[2019\]](#). The normalization placement analysis of [Xiong et al. \[2020\]](#) provides relevant background for interpreting depth sweeps without turning initialization theory into a performance guarantee.

Section IV

Categorical Models of Causality

Adjoint Functors

An adjunction is a universal rule for translating maps. It does not usually make two categories isomorphic, nor does it say that two processes are causal duals. Instead, it says that a construction performed on one side solves a mapping problem on the other side in a way that is natural in both inputs. This distinction matters in AI: a suggestive translation between model classes is not an adjunction until its categories, morphisms, and universal property have all been specified.

Adjunctions organize free constructions, abstraction and concretization, limits and colimits, monads, and Kan extensions. These uses recur throughout the remainder of the book. This chapter develops the common formal core and then explains the additional work required before applying it to causal models.

The Hom-Set Definition

Let $F : \mathcal{C} \rightarrow \mathcal{D}$ and $G : \mathcal{D} \rightarrow \mathcal{C}$ be functors.

Definition 46 (Adjunction). *An adjunction $F \dashv G$ is a family of bijections*

$$\Phi_{c,d} : \mathcal{D}(Fc, d) \cong \mathcal{C}(c, Gd), \quad (34)$$

natural in every $c \in \mathcal{C}$ and $d \in \mathcal{D}$. The functor F is the left adjoint; G is the right adjoint.

If $u : Fc \rightarrow d$, write $u^\sharp = \Phi_{c,d}(u)$. If $v : c \rightarrow Gd$, write $v^\flat = \Phi_{c,d}^{-1}(v)$. Naturality says that for $h : c' \rightarrow c$ and $k : d \rightarrow d'$,

$$(k \circ u)^\sharp = Gk \circ u^\sharp, \quad (u \circ Fh)^\sharp = u^\sharp \circ h. \quad (35)$$

The first equation is covariance in d ; the second is contravariance in c . No informal appeal to translation is enough: these two equations are part of the claim.

The turnstile notation records the direction of the two functors:

$$\mathcal{C} \begin{array}{c} \xrightarrow{F} \\ \dashv \\ \xleftarrow{G} \end{array} \mathcal{D}$$

It does not assert that F and G are inverses.

The Free-Monoid Example

The canonical example explains the words *free* and *forgetful*. Let $U : \mathbf{Mon} \rightarrow \mathbf{Set}$ forget multiplication, unit, and laws, retaining only the underlying set. Let $F : \mathbf{Set} \rightarrow \mathbf{Mon}$ send a set X to the free monoid X^* of finite words in X . Every function $a : X \rightarrow U(M)$ extends uniquely to a monoid homomorphism

$$\hat{a} : X^* \longrightarrow M, \quad \hat{a}(x_1 \cdots x_n) = a(x_1) \cdots a(x_n).$$

Restriction to one-letter words recovers a . Hence

$$\mathbf{Mon}(FX, M) \cong \mathbf{Set}(X, UM), \quad (36)$$

naturally in X and M , and $F \dashv U$.

This example also supplies a warning. Calling a functor “forgetful” does not prove that it has a free left adjoint. The proposed free object must exist, and maps from it must satisfy the unique-extension property in Equation (36).

Unit, Counit, and Triangle Identities

The hom-set bijection packages two canonical natural transformations. The *unit* and *counit* are

$$\eta_c = (1_{Fc})^\sharp : c \rightarrow GFc, \quad \varepsilon_d = (1_{Gd})^\flat : FGd \rightarrow d. \quad (37)$$

Their components obey the triangle identities

$$\varepsilon_{Fc} \circ F\eta_c = 1_{Fc}, \quad G\varepsilon_d \circ \eta_{Gd} = 1_{Gd}. \quad (38)$$

Theorem 3. *Giving an adjunction by the natural bijections in Equation (34) is equivalent to giving functors F, G , a unit $\eta : 1_C \Rightarrow GF$, and a counit $\varepsilon : FG \Rightarrow 1_D$ satisfying Equation (38).*

Proof. The bijection determines η and ε by Equation (37). Applying each transpose and then its inverse to an identity gives the triangle equations. Conversely, define

$$u^\sharp = Gu \circ \eta_c, \quad v^\flat = \varepsilon_d \circ Fv.$$

The triangle identities show that these operations are mutually inverse, and naturality of η and ε gives naturality in both variables. $\square$

For the free-monoid adjunction, $\eta_X : X \rightarrow U(X^*)$ inserts a generator as a one-letter word. The counit $\varepsilon_M : F(UM) \rightarrow M$ evaluates a formal word by multiplying its letters in M . The first operation introduces syntax; the second interprets that syntax in a model.

Galois Connections

A preorder is a category with at most one morphism between any two objects. For monotone maps $f : P \rightarrow Q$ and $g : Q \rightarrow P$, the adjunction condition reduces to

$$f(p) \leq_Q q \iff p \leq_P g(q). \quad (39)$$

Such an adjunction is a *Galois connection*. It is useful for abstraction: f may construct an abstract description and g may return the most general concrete property licensed by that description.

Theorem 4. *If $f \dashv g$ is a Galois connection, then f preserves every join that exists in P , and g preserves every meet that exists in Q .*

Proof. Let $A \subseteq Q$ have meet $m = \bigwedge A$. For $p \in P$,

$$p \leq g(m) \iff f(p) \leq m \iff (\forall a \in A) f(p) \leq a \iff (\forall a \in A) p \leq g(a).$$

Thus $g(m)$ has exactly the universal property of $\bigwedge_{a \in A} g(a)$. The join statement is dual. $\square$

This proof is more informative than a slogan: preservation follows from the hom-order equivalence, not merely from monotonicity.

Limits and Colimits as Adjoints

Let J be a small category and let $\Delta : \mathcal{C} \rightarrow \mathcal{C}^J$ send c to the constant diagram at c . Whenever all J -shaped limits and colimits exist,

$$\text{colim} \dashv \Delta \dashv \text{lim}. \quad (40)$$

Indeed,

$$\mathcal{C}(\text{colim } D, c) \cong \mathcal{C}^J(D, \Delta c), \quad \mathcal{C}^J(\Delta c, D) \cong \mathcal{C}(c, \text{lim } D).$$

The left bijection is the universal property of a cocone; the right is the universal property of a cone.

Theorem 5. *Every right adjoint preserves all limits that exist, and every left adjoint preserves all colimits that exist.*

The theorem says “preserves,” not “creates.” If a limit exists in the source category, a right adjoint carries its limiting cone to a limiting cone. It does not guarantee that the source limit exists. The same qualification applies dually to colimits.

Adjunctions, Monads, and Equivalences

Every adjunction $F \dashv G$ induces a monad on $\mathcal{C}$:

$$T = GF, \quad \eta : 1_{\mathcal{C}} \Rightarrow T, \quad \mu = G\epsilon F : T^2 \Rightarrow T. \quad (41)$$

The triangle identities imply the monad laws. Probability can be represented by particular monads, but Equation (41) does not make an arbitrary adjunction probabilistic; the carrier category and probability construction must be specified separately.

Nor is an adjunction normally an equivalence. It becomes an equivalence when its unit and counit are natural isomorphisms. Thus *left/right*, *free/forgetful*, *syntax/semantics*, and *observation/intervention* are not interchangeable pairs. Each proposed interpretation needs its own construction and proof.

What an Adjoint Claim Would Require in Causality

Conditional-independence information does not generally determine a unique causal directed acyclic graph. For example, the chains

$$X \longrightarrow Y \longrightarrow Z \quad \text{and} \quad X \longleftarrow Y \longleftarrow Z$$

have the same skeleton and entail the same graphical conditional independence $X \perp Z \mid Y$. Distinguishing causal direction requires additional assumptions, environments, temporal information, or interventions.

It is tempting to describe the passage from a causal model to its observational independences as “forgetful” and the reverse passage as “free.” That may be a productive research proposal, but the adjectives do not supply an adjunction. A valid theorem would need:

1. a category of observational structures and its morphisms;
2. a category of causal models and its morphisms;
3. a functor U that extracts the declared observational structure;
4. a functor F that constructs a causal candidate; and
5. a natural bijection $\mathbf{Causal}(FS, M) \cong \mathbf{Obs}(S, UM)$.

Even if such an adjunction exists for chosen categories, it certifies a universal completion relative to those definitions. It does not by itself certify causal identification, empirical truth, or the semantics of an intervention.

This boundary clarifies the role of the next chapter. DEMOCRITUS extracts candidate causal claims and organizes them into inspectable models [Mahadevan, 2025d]. That engineering map is not called an adjunction here. Its outputs still require provenance, consistency checks, and independent causal evidence.

Summary

An adjunction is a natural family of map transpositions. Its unit and counit encode universal introduction and interpretation operations; its triangle identities ensure that the two transpositions agree. Free constructions, Galois connections, limits, colimits, and monads are instances of this one pattern.

For AI, the main lesson is methodological. Adjoint language becomes valuable when it exposes and proves a universal mapping property. Without that proof, “free,” “forgetful,” and “dual” remain analogies. Maintaining this distinction will be especially important when the following chapters move from statistical evidence to causal structure.

Suggested Reading

Adjunctions, units, counits, and the adjoint functor theorems are treated systematically by [Mac Lane \[1971\]](#) and [Riehl \[2017\]](#). [Fong and Spivak \[2018\]](#) develops Galois connections and other applied examples. Kan’s original paper remains foundational for adjoint functors and extensions [[Kan, 1958](#)]; [van Belle \[2024\]](#) develops Kan extensions in probability while carefully separating the categorical construction from its probabilistic interpretation.

Causal Claims from Language

DEMOCRITUS compiles documents into inspectable collections of causal-form claims and local graph neighborhoods [Mahadevan, 2025d]. A revised version of the DEMOCRITUS system adds a post-extraction layer that normalizes causal mentions, localizes them at declared weak equivalences, diagnoses regime-sensitive failures of gluing, and preserves the resulting decisions in auditable database artifacts [Mahadevan, 2026d]. Its distinctive aim is breadth: rather than begin with a small prespecified causal graph, it uses a document to seed a much wider frame of discourse. This makes the system useful for hypothesis generation and causal auditing. It does *not*, by itself, identify causal effects from observational text.

Boundary: Interpretive boundary

The implemented pipeline answers the question “Which causal-form relations and graph neighborhoods were compiled from this discourse?” It does not answer, without additional evidence, “Which of these relations are true causal mechanisms?” Extraction, ranking, identification, and validation are separate stages and must not be collapsed.

From a Document to a Finite Causal Presentation

Let D be a document and let θ record the prompts, model, decoding configuration, and software revision used for compilation. The front end produces a finite artifact

$$T_\theta(D) = \{(s_i, r_i, o_i, t_i)\}_{i=1}^N,$$

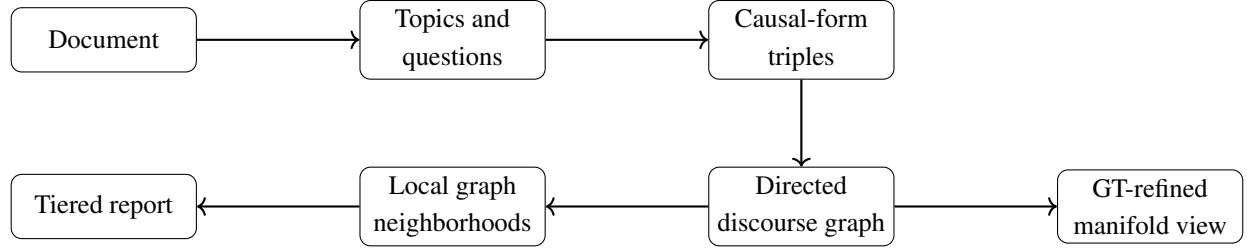
where s_i and o_i are extracted entities, r_i is a relation phrase, and t_i records the topic or question that elicited the statement. A production artifact should additionally retain the source span, whether the statement was quoted, retrieved, or generated, and the complete generator provenance.

The system workflow can be summarized as follows.

Write $Q(T)$ for the directed graph obtained from the triples. The local objects assembled for reporting are bounded directed ego neighborhoods,

$$L_{v,\rho,m} = \text{cap}_m(\text{Ego}_\rho^\rightarrow(Q(T), v)),$$

centered at a focus node v , with radius ρ and node cap m . Calling this ob-



ject a “causal model” describes its intended interpretation; formally, it is a subgraph of a graph of linguistic claims.

The Typed Presentation

The pipeline becomes clearer when represented as a finite sketch. Useful object types include

Document, Topic, Question, Statement, Entity, Claim.

Arrows record such operations as topic discovery, question generation, statement extraction, and the source and target of a claim. Path equations can require, for example, that every claim point back to the statement and document from which it arose. An auditable extension should add *SourceSpan*, *GeneratorRun*, *Evidenceltem*, and *Intervention* rather than storing every relation in one undifferentiated edge table.

This sketch presentation is useful even though the implemented compiler has not been shown to be a left adjoint. To establish an adjunction between a category of discourse presentations and a category of causal models, one must define both categories, both functors, and a natural hom-set bijection. A round-trip desideratum—formalize a passage and then explain the result—is not by itself a unit, counit, or proof of adjointness. The adjunction remains a design hypothesis whose exact theorem contract was set out in the preceding chapter.

Relational Geometry with the Geometric Transformer

A sentence encoder first assigns an initial vector $h_v^{(0)}$ to every entity v in the discourse graph. The Geometric Transformer then refines these vectors by propagating directed, relation-sensitive messages. If an edge $u \rightarrow v$ carries relation type $r(u, v)$, one layer has the schematic form

$$m_{u \rightarrow v}^{(\ell)} = M_\ell \left[h_u^{(\ell)}, h_v^{(\ell)}, e_{r(u,v)} \right], \quad h_v^{(\ell+1)} = h_v^{(\ell)} + \sum_{u \rightarrow v} m_{u \rightarrow v}^{(\ell)}$$

After two such layers and a final normalization, UMAP projects the refined vectors to two or three dimensions. Domain-coherent length-two paths are

Figure 33: The compilation core of DEMOCRITUS has two complementary geometric views. One extracts and ranks local graph neighborhoods; the other refines the relational representation with a Geometric Transformer (GT) before manifold projection. The post-extraction layer subsequently normalizes and localizes causal mentions and tests whether document-local claims glue across regimes. Extraction, geometry, and post-extraction coherence require distinct evaluation criteria.

retained as 2-simplices, so the resulting representation records more than an untyped cloud of entity embeddings.

The geometric construction must nevertheless be interpreted at the level at which it is realized. The manifold path evaluates a newly instantiated GT without loading learned parameters, and it does not optimize a Diagrammatic Backpropagation objective. It therefore implements a relation-aware geometric refinement, but not a trained GT+DB model. Architecture, learning rule, and semantic validity are three distinct claims.

Experiment: Democritus GT rendering ablation

An ablation in the DEMOCRITUS study compares two renderings of the same economics corpus of approximately 35,000 causal variables [Mahadevan, 2025d]. The first baseline applies UMAP directly to the sentence-encoder representations of the extracted relational triples; the resulting projection is visually unstructured. The second baseline inserts GT refinement before the same projection step; the rendered manifold then displays clearly separated structures associated with economic subfields. This side-by-side experiment is evidence that GT refinement materially improved the *rendered organization* of that LCM. It is a qualitative visualization ablation, however, rather than a calibrated measure of causal validity or a test of the independent neighborhood-ranking branch considered below.

Ranking Local Neighborhoods

The second view ranks local discourse neighborhoods. For a local graph $L = (V_L, E_L)$, the implemented score is

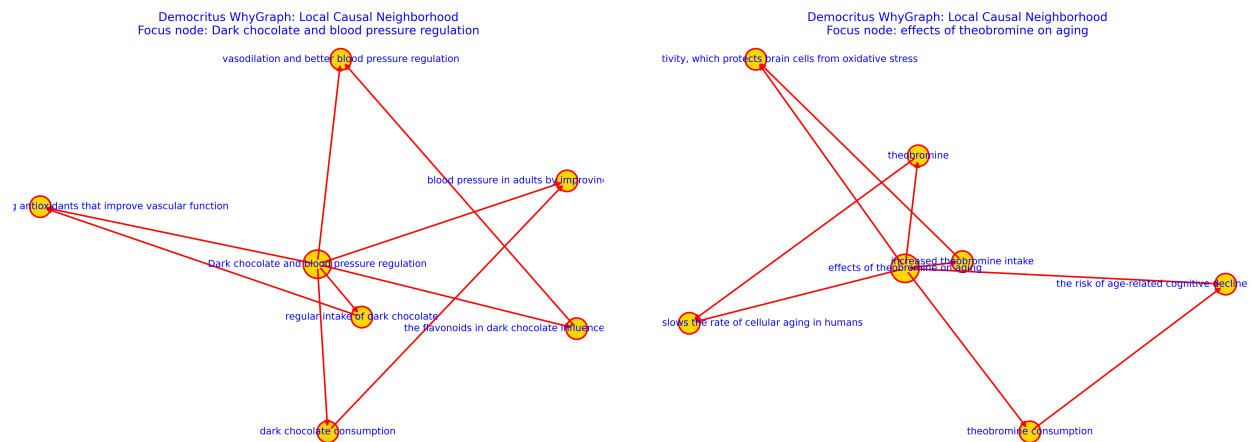
$$S(L; T) = \sum_{(u,v) \in E_L} \log(1 + n_T(u, v)) - \lambda_{\text{edge}} |E_L|,$$

where $n_T(u, v)$ counts occurrences of the ordered endpoint pair in the compiled triple file. The evaluated configuration uses $\lambda_{\text{edge}} = 0.25$. Relation labels do not enter this support count. With a single occurrence per edge, each edge therefore contributes $\log 2 - 0.25 \approx 0.443$.

The top- K neighborhood scores are converted to softmax weights

$$w_j = \frac{\exp S(L_j)}{\sum_{k=1}^K \exp S(L_k)}.$$

An edge receives a report value by summing the weights of selected neighborhoods that contain it. This is a deterministic weighted-overlap statistic. It is not a calibrated posterior probability, and the overlap does not constitute agreement among independent causal models: the neighborhoods are extracted from the same compiled graph and commonly overlap by construction.



Boundary: What each observable does—and does not—establish	
Observable	What is not yet licensed
Causal-form subject–relation–object triple	Truth, direction, or an intervention semantics
Repeated generated endpoint pair	Independent evidence or source corroboration
Large, dense local neighborhood	Semantic plausibility or causal importance
Top- <i>K</i> weighted overlap	A Bayesian posterior or expert consensus
Sparse high tier	Calibrated abstention

Figure 34: Two discourse neighborhoods for the dark-chocolate example. The left graph has 17 edges and score 7.533; the right has 9 edges and score 3.988. Under the neighborhood-scoring rule, most of this difference follows directly from graph size. The comparison therefore describes the compiled discourse geometry; it does not adjudicate whether cholesterol or theobromine is the medically correct mechanism.

A Running Example: Dark Chocolate and Aging

Consider a *Washington Post* article about dark chocolate, theobromine, and biological aging. A conventional summary properly foregrounds the article’s principal association and caveats. The discourse compiler instead expands the topic frame and extracts graph neighborhoods about aging, methylation, cardiovascular markers, diet, and related concepts.

This example exposes both the value and the limitation of the method. It can show that a narratively central claim occupies a relatively sparse part of an expanded discourse graph, while a different family of relations is densely represented. That is a useful auditing observation. But generated redundancy is not evidence multiplication: asking one model for several paraphrases cannot substitute for independent studies, interventions, or expert appraisal.

An Empirical Audit across Four Domains

The table below summarizes quantities computed directly from the system’s triple files, neighborhood scores, and generated reports. “Unique graphs” counts distinct canonical graph payloads rather than configurations, since many focus–radius–cap settings produce the same subgraph.

These distinctions matter when interpreting the table. The 80 Antarctica records are scored configurations rather than approximately 3,000 distinct local models, and the reduction from configurations to a top- K report is a selection operation rather than a measured compression ratio. A triple file alone is also insufficient for reproducibility: the code revision, prompts, model identifier, decoding settings, random seeds, source document, and all post-processing configurations are part of the experimental object. The available batch manifests are incomplete and, for some displayed domains, do not provide a single joined record from source document to final report.

External Calibration against UniCausal

The later DEMOCRITUS system separates evaluation of the upstream causal-relation extractor from evaluation of the subsequent localization and gluing layers [Mahadevan, 2026d]. The extraction comparison uses the grouped AltLex test split distributed with UniCausal [Hidey and McKeown, 2016, Tan et al., 2023]. Grouping annotations by passage yields 404 passages—115 causal and 289 non-causal—containing 127 annotated directed cause–effect pairs. Every method receives the same untagged text. The restricted DEMOCRITUS adapter uses a deterministic JSON contract and accepts only cause and effect spans copied verbatim from the passage; this benchmark bypasses retrieval, topic expansion, localization, and gluing.

Table 16: Descriptive statistics for four DEMOCRITUS domains. $r(S, |E|)$ is the Pearson correlation between score and edge count. The near-perfect correlations are expected from the scoring formula and show why the score must not be interpreted as an independently validated measure of semantic plausibility.

Domain	Triples	Scored	Unique	Max	$r(S, E)$	Tier 1/2/3
Antarctica	283	80	60	6.647	1.000	11/2/11
Dark chocolate	336	3,000	280	7.534	.999	16/1/20
Indus Valley	251	2,985	224	3.102	1.000	3/0/13
Dinosaur extinction	293	3,000	282	10.155	.999	22/0/3

Method	Cls. P	Cls. R	Cls. F1	Cause F1	Effect F1	Span F1	Exact pair F1	Relaxed pair F1
Cue-pattern baseline	50.0	22.6	31.1	—	—	—	—	—
UniCausal public baseline	52.0	68.7	59.2	40.8	46.5	43.6	20.5	37.3
DEMOCRITUS, restricted Kimi K2.5	44.1	80.9	57.1	37.1	41.1	39.1	0.0	35.8

Experiment: Democritus–UniCausal extraction comparison

UniCausal is stronger overall on this benchmark: its classification F1 is 59.2% rather than 57.1%, and its macro span F1 is 43.6% rather than 39.1%. DEMOCRITUS instead occupies a higher-recall, lower-precision operating point, finding 80.9% of causal passages at 44.1% precision, compared with 68.7% recall at 52.0% precision for UniCausal. The zero exact-pair F1 is an important boundary rather than a number to hide: inspection found that the adapter often selected a semantically appropriate but broader clause than the compact annotated argument. Its 35.8% relaxed-pair F1 therefore diagnoses useful directional overlap together with poor boundary calibration.

This experiment does not show that the categorical machinery improves sentence-level extraction. It shows that the upstream interface is externally comparable and identifies a concrete precision–recall tradeoff. The role of the categorical layer begins after candidate extraction: it declares which normalization-preserving rewrites may be inverted, keeps polarity, temporal scope, and regime distinctions from being collapsed, and records where local claim families fail to assemble coherently. Evidence for the later merge and gluing decisions comes from system traces and representative claims rather than an independent expert-annotated benchmark.

Table 17: Upstream extraction comparison on 404 grouped AltLex/UniCausal passages. Values are percentages. The table evaluates causal-passage classification and span recovery, not the post-extraction localization and gluing layers.

Geometry and Completion: A Boundary on the Claims

The two views answer different questions. GT refinement organizes the global geometry of a claim graph, whereas the ranking rule orders bounded local neighborhoods. The local scores reported above therefore neither measure nor ablate the contribution of the GT. The DEMOCRITUS study instead evaluates GT refinement on an economics LCM with roughly 35,000 variables. Its GT-refined projection is substantially more organized than UMAP applied directly to the initial embeddings. This establishes a qualitative visualization advantage in the reported setting. Repeated-seed geometric metrics and held-out downstream tasks would be needed to support a broader quantitative claim about representation quality or causal utility.

Likewise, the implemented neighborhood sweep is not iterative horn filling, simplicial completion, or sheaf repair. Those ideas suggest valuable future algorithms: one might detect a missing compositional relation, propose

a typed completion, and test it against held-out evidence. But a graph ego operation only selects existing vertices and edges; it neither invents a missing edge nor repairs incompatible local sections. Distinguishing an implemented operator from a motivating analogy is part of categorical discipline.

An Admission Protocol for Causal Use

DEMOCRITUS is best viewed as a candidate generator and structural index. Moving from discourse geometry to causal knowledge requires an admission firewall.

1. **Freeze provenance.** Preserve source spans, retrieval records, generator runs, prompts, configurations, and code revisions.
2. **Separate evidence types.** Never pool quoted source claims, retrieved claims, and model-generated hypotheses as interchangeable support.
3. **Evaluate extraction.** On an expert-annotated corpus, report entity, relation, direction, negation, modality, and provenance errors.
4. **Use independent ranking evidence.** Compare graph-only ranking with source entailment, bibliographic independence, study quality, and expert judgments; include simple degree and frequency baselines.
5. **Ablate every module.** Topic expansion, graph construction, geometry, scoring, and report reconstruction need separate controls.
6. **Require causal admission.** A claim intended for decision making must state its causal model, identification assumptions, intervention target, estimand, and empirical validation route. Otherwise it remains a discourse hypothesis.

This protocol does not diminish the contribution. It locates it more precisely. Large language models make it possible to construct broad, inspectable atlases of causal discourse; categorical presentations make the types, maps, and integrity constraints explicit. The resulting artifact can help experts find claims, compare local explanations, and identify evidential gaps. Causal identification begins where compilation ends.

Summary

DEMOCRITUS turns documents into finite presentations of causal-form claims and organizes them as directed discourse graphs. It provides GT-refined manifold projections and bounded local neighborhoods, then normalizes extracted mentions into claim classes whose compatibility can be examined across documents and regimes. The local scoring rule is deterministic after LLM compilation and unusually easy to audit; the manifold path

performs relation-aware refinement without learned GT parameters. Precisely because the system is inspectable, we can state its limits sharply: the neighborhood score is dominated by edge count, overlap is not independent corroboration, and neither branch by itself identifies interventions or causal effects. These boundaries turn vague claims about “causality from language” into a concrete research program for moving from causal claims in language to admitted causal knowledge.

Suggested Reading

The distinction between associational, interventional, and counterfactual questions is developed by [Pearl \[2009\]](#); causal discovery assumptions are surveyed by [Zanga and Stella \[2023\]](#). The original large-causal-model system is described by [Mahadevan \[2025d\]](#). For categorical approaches to linguistic and semantic structure, see [Bradley et al. \[2021\]](#). The homotopy-localized extension and its UniCausal comparison are given by [Mahadevan \[2026d\]](#); the UniCausal benchmark and public models are described by [Tan et al. \[2023\]](#). PROMETHEUS extends document-centered extraction into a causal research workflow [[Mahadevan, 2026h](#)], while ODYSSEY studies verifiable foundries that preserve provenance and validation boundaries [[Mahadevan, 2026g](#)].

Temporal Diffusion over Causal Trajectories

Modeling Corporate Geometry

The previous chapter developed DEMOCRITUS as a pipeline for compiling causal-form claims into structured discourse graphs. There the emphasis was primarily local: given a corpus of documents, extract an auditable state capturing relations asserted or generated around the text. In this chapter we move from isolated causal snapshots to *temporal corporate geometry*. The object of study is no longer a single causal graph, but a sequence of yearly causal states extracted from a firm’s disclosures and then repaired into a coherent trajectory through time.

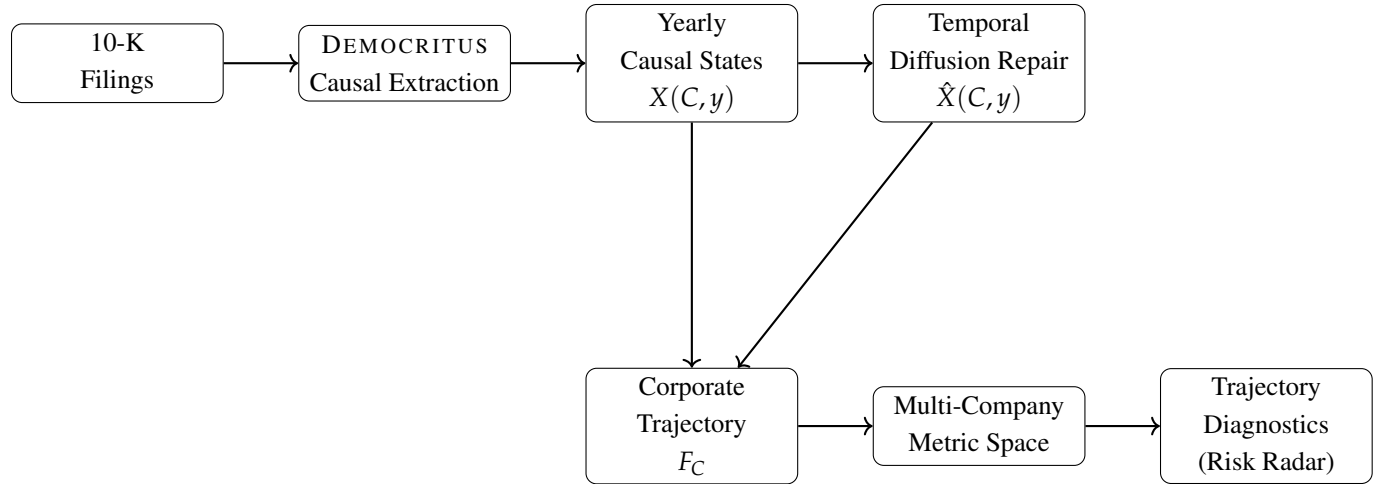
This shift mirrors a broader theme developed earlier in the book. A raw yearly extraction is not yet a finished semantic object. It is better understood as a noisy, partially specified structured state. The temporal problem is therefore not merely to compare adjacent years, but to test whether a learned regularizer can reconstruct deliberately corrupted sequences of partial discourse states. In this sense, temporal diffusion over corporate states plays a role analogous to denoising in structured language modeling: it proposes a global representation from a sequence of local, noisy objects. Whether that proposal is more faithful than simpler baselines is an empirical question, not a consequence of the categorical notation.

We refer to this problem as *modeling corporate geometry*. The term is meant literally. A corporation is represented not only by text, or by static financial ratios, but by a path through a space of causal configurations. Once such paths are available, one can ask geometric questions: Which firms follow similar trajectories? When does a firm’s path become unstable? Which years represent smooth continuation, and which correspond to structural reorganization or stress?

Yearly Corporate States as Causal Objects

For each company C and fiscal year y , the brand-aware DEMOCRITUS pipeline constructs a causal state

$$X(C, y),$$



summarizing the causal-form relations compiled from the firm’s textual disclosures. The adjective *causal* refers to the semantics of the extracted claims. It does not mean that those relations have been identified from interventions or certified as causal mechanisms. These relations may involve themes such as

- innovation and research activity,
- pricing power and demand growth,
- inventory and supply-chain constraints,
- margin pressure, regulation, and operational risk.

Each $X(C, y)$ is therefore a structured object rather than a simple feature vector. It records a pattern of dependencies among business drivers and risks, and may be viewed as a point in a high-dimensional *causal state space*. But these points are noisy: they are extracted from text, shaped by reporting conventions, and subject to both semantic ambiguity and year-to-year extraction error.

For this reason, the yearly states should not be treated as perfectly observed ground truth. They are better viewed as partially reliable observations of an underlying corporate process.

Corporate Trajectories as Functors Through Time

A company evolves. Its products, constraints, markets, and strategic posture all change, and the corresponding causal structure changes with them. Thus the sequence

$$X(C, y_0), X(C, y_1), X(C, y_2), \dots$$

is naturally interpreted as a trajectory through causal state space.

Figure 35: Corporate geometry pipeline. Textual evidence from corporate filings is transformed into yearly causal states using the DEMOCRITUS pipeline. The temporal model maps these states to candidate denoised trajectories. The resulting representations support descriptive cross-company comparisons and a heuristic structural-indication dashboard; neither output is, by itself, a validated causal model or a predictor of corporate survival.

Categorically, this trajectory may be represented as a functor

$$F_C : \mathbf{Year} \longrightarrow \mathbf{CausalState},$$

where $F_C(y) = X(C, y)$. The objects of **Year** are fiscal years, and the morphisms encode temporal progression. The image of this functor is a time-indexed sequence of causal states for company C .

This is the basic geometric move of the chapter: a firm's disclosures are lifted from a sequence of documents to a sequence of structured causal states, and that sequence is then treated as a path in a space of corporate semantics. Once corporate history is expressed in this way, one can compare companies by comparing the shapes of their trajectories rather than only their annual statistics.

Why Geometry Matters

Viewing firms geometrically has several advantages.

- It allows companies to be compared by distances between their causal trajectories rather than by isolated yearly features.
- Structural changes in strategy, operations, or risk appear as movements in causal state space.
- Instabilities in a trajectory can signal periods of transition, fragility, or strategic stress.
- Firms in related sectors often follow nearby paths, revealing shared geometric structure at the industry level.

At the same time, naive trajectories are unreliable. Year-to-year differences may reflect genuine change, but they may also reflect extraction noise, changing disclosure style, or accidental variation in the textual evidence. A central challenge is therefore to distinguish meaningful temporal movement from artifact.

Temporal Diffusion as Structured Repair

This motivates the temporal diffusion framework of the chapter. Rather than treating the extracted yearly states $X(C, y)$ as final, we view them as noisy observations of a latent temporal process and seek repaired states

$$\hat{X}(C, y)$$

that are both locally faithful and temporally coherent.

The word *diffusion* is important here. The goal is not simply to smooth trajectories in an ad hoc way. Rather, we seek a structured repair process that propagates compatibility across time while preserving genuine corporate

change. In the language developed earlier in the book, this is closer to a completion problem than to a static embedding problem: each year constrains its neighbors, and the repaired trajectory should become a coherent temporal object rather than a sequence of unrelated snapshots.

This candidate trajectory then supports two downstream constructions. First, it enables the construction of multi-company metric spaces in which firms may be compared by causal-geometric similarity. Second, it yields descriptive signals summarized by an instability or structural-indication radar. The reported experiments do not establish these scores as early warnings of financial distress, failure, or survival.

From Causal Snapshots to Corporate Geometry

The previous DEMOCRITUS chapter showed how to construct individual causal states from text. The goal of the present chapter is to connect those states into a temporal geometry.

The progression of ideas is therefore:

$$\text{text} \longrightarrow X(C, y) \longrightarrow \hat{X}(C, y) \longrightarrow F_C$$

from raw textual evidence, to yearly causal states, to repaired temporal states, and finally to full corporate trajectories. What emerges is a new representation of the firm: not as a static balance sheet or a bag of words, but as a path through a space of evolving causal structure.

The rest of the chapter develops the mathematical and algorithmic tools for constructing these repaired trajectories, defining distances between them, and using them to analyze the geometry of corporate evolution.

From Static Causal Snapshots to Temporal Structure

For each company C and fiscal year y , the brand-aware DEMOCRITUS pipeline produces a structured causal state

$$X(C, y),$$

summarizing the causal relations, motifs, latent summaries, and regime indicators extracted from that year's disclosures. Each $X(C, y)$ is therefore a rich structured object rather than a simple feature vector. It captures a snapshot of the firm's inferred causal organization at a particular moment in time.

Taken across years, these snapshots form a sequence

$$X(C, y_0), X(C, y_1), X(C, y_2), \dots$$

which may be interpreted as a trajectory of corporate causal states. Categorically, this trajectory is naturally represented as a functor

$$F_C : \mathbf{Year} \longrightarrow \mathbf{CausalState},$$

mapping each fiscal year to its corresponding causal object.

The earlier DEMOCRITUS chapter therefore supplied the *objects* of a temporal causal system. What remains is to understand the *morphological evolution* of those objects through time. How does a firm move from one causal configuration to the next? Which apparent changes reflect genuine strategic reorganization, and which are artifacts of extraction noise, reporting style, or transient textual fluctuations?

This chapter addresses that temporal question directly.

Yearly Snapshots Are Partial and Noisy

Although the yearly states $X(C, y)$ are highly structured, they should not be treated as perfectly observed ground truth. Several sources of variability enter the extraction process:

- language-model extraction noise,
- transient motifs that appear in only one year,
- missing relations due to incomplete textual evidence,
- abrupt differences caused by reporting style rather than real structural change.

As a result, the raw year-to-year sequence often contains irregular jumps, spurious motifs, or missing links. The observed trajectory

$$X(C, y) \rightarrow X(C, y + 1)$$

may therefore reflect a mixture of genuine corporate evolution and measurement noise.

From the perspective developed in the previous chapters, this suggests that a yearly causal state is better viewed as a *partially reliable structured object*. The temporal problem is then not merely to model transitions directly, but to *repair* a sequence of such objects into a more coherent trajectory.

Temporal Blocks as Local Diagrams

To do this, we work locally in time. For a company C and starting year y , define a temporal block

$$B(C, y) = (X(C, y), X(C, y + 1), \dots, X(C, y + k)),$$

where k is the block length minus one.

A temporal block is more than a tuple of yearly states. It is a local diagram in the category of causal states: a small time-indexed configuration whose objects are yearly causal snapshots and whose implicit morphisms represent their temporal relations. Noise may corrupt any part of this diagram. Edges

may disappear, motifs may drift, latent summaries may fluctuate, and one anomalous year may distort the apparent local evolution.

The task of the temporal model is therefore not simply to smooth scalar time series. It is to reconstruct the most coherent local diagram compatible with the available temporal evidence.

Temporal Diffusion as Structured Repair

This leads to the central construction of the chapter. Rather than modeling yearly transitions directly, we introduce a corruption-and-repair process over temporal blocks.

A clean block

$$B(C, y)$$

is transformed into a corrupted block

$$\tilde{B}(C, y),$$

and a learned deterministic denoising model returns

$$\hat{B}(C, y) = f_{\theta}(\tilde{B}(C, y), t).$$

The resulting model does not attempt to generate corporate trajectories from scratch. Instead, it solves a *structured completion problem* over time: given a noisy local diagram, reconstruct a temporally coherent one. This is the corporate analogue of the denoising regimes studied in the previous chapter on structured language modeling. There, denoising was posed as completion from partial context rather than one-shot generation. The same design principle motivates the present model; whether it yields better reconstruction must be decided by the holdout tests below.

Repeated over overlapping temporal windows, this repair process yields a new sequence of states

$$\hat{X}(C, y),$$

and hence a repaired functor

$$\hat{F}_C : \mathbf{Year} \longrightarrow \mathbf{CausalState}.$$

The aim is not to erase real corporate change, but to suppress local artifacts and strengthen temporal coherence. In this sense, temporal diffusion acts as a repair operator on company functors.

A Right-Kan Intuition

The categorical intuition behind this repair process is also worth making explicit. In earlier chapters, left Kan extensions provided a natural language for aggregation from local evidence. Temporal diffusion suggests the complementary viewpoint: each repaired state should be compatible with multiple

local temporal constraints at once. A year must agree not only with its own extracted evidence, but also with neighboring years and with the broader shape of the local block.

This suggests a *right-Kan-style* analogy, although the implementation does not compute a Kan extension. The problem is not merely to pool information, but to construct a state that satisfies as many relevant compatibility conditions as possible. We do not require the full formal machinery here, but the intuition is important: corporate trajectory repair is closer to structured completion under temporal constraints than to ordinary smoothing.

From Individual Trajectories to a Company Metric Space

Once repaired trajectories are available, one can compare firms at the level of temporal causal geometry rather than isolated yearly features. Given two companies C and D , we may seek yearwise comparison maps

$$\eta_{C,D,y} : X(C, y) \longrightarrow X(D, y),$$

or, more generally, comparisons between their repaired trajectories $\hat{F}_C$ and $\hat{F}_D$.

Viewed categorically, such comparisons resemble natural transformations between company functors. The extent to which the corresponding local diagrams align provides a measure of similarity between corporate dynamics. This leads naturally to a *multi-company metric space* whose points are repaired causal trajectories rather than static corporate profiles.

The rest of the chapter develops this temporal diffusion framework in detail and then applies it to preliminary multi-company case studies. The central goal is to define the proposed repair and comparison operators, then determine from preserved artifacts and holdout tests which claims about their empirical behavior are presently justified.

Temporal Block Denoising Architectures

The temporal diffusion model extends the denoising viewpoint developed earlier for structured language modeling and relational-triple repair. Rather than repairing yearly causal states independently, it operates on *temporal blocks*: short windows of consecutive yearly causal objects. The goal is not merely to smooth these windows numerically, but to repair them as local temporal diagrams.

Temporal Block Representation

Let C denote a company and y a fiscal year. For each year, the brand-aware DEMOCRITUS pipeline produces a structured causal state

$$X(C, y) \in \mathbf{CausalState}.$$

A temporal block of length $k + 1$ is then

$$B(C, y) = (X(C, y), X(C, y + 1), \dots, X(C, y + k)).$$

In the reported implementation, each yearly state $X(C, y)$ is encoded as a signed, normalized vector over a canonical edge or motif basis. The model also receives total mass, nonzero-edge count, low-rank latent coordinates, transition summaries, fiscal-year coordinates, block position, and corruption timestep. Thus the input lies in $\mathbb{R}^d$, but it is a finite numerical encoding of a discourse graph rather than the graph itself. Related designs could additionally include

- edge-weight vectors over a canonical relation basis,
- motif counts and higher-order interaction statistics,
- regime indicators and stability flags,
- latent embeddings summarizing the yearly causal graph.

Stacking these yearly representations yields a block tensor

$$B(C, y) \in \mathbb{R}^{(k+1) \times d}.$$

This tensor should be interpreted as a local temporal diagram rather than a flat matrix: each row corresponds to a yearly causal object, and neighboring rows encode short-range temporal relationships.

Temporal Corruption Model

To train a temporal denoiser, we introduce a corruption operator

$$\mathcal{N} : \mathbf{Block} \rightarrow \widetilde{\mathbf{Block}},$$

which maps a clean temporal block $B(C, y)$ to a corrupted block

$$\tilde{B}(C, y) = \mathcal{N}(B(C, y)).$$

The implemented corruption process is synthetic. It is intended to resemble plausible extraction defects, but the frequencies of those defects have not been estimated from an independently annotated filing corpus. Its perturbations include:

- **edge dropout:** randomly remove inferred causal relations,
- **sign flips and jitter:** reverse or perturb selected weights,
- **spike insertion:** add small mass at previously zero coordinates,
- **temporal dropout and transfer:** remove a salient middle-year coordinate or move part of its mass to an adjacent year.

More explicitly, one may view the corruption process as a composition of independent stochastic operators

$$\mathcal{N} = \mathcal{N}_{\text{drop}} \circ \mathcal{N}_{\text{flip/spike}} \circ \mathcal{N}_{\text{weight}} \circ \mathcal{N}_{\text{temporal}},$$

with each component activated according to tunable probabilities. These probabilities define the *temporal partial-information regime*: they determine what kinds of local temporal defects the model is trained to repair.

This is the direct analogue of the corruption operator in denoising language modeling. There, a future token block was partially corrupted. Here, a local time-indexed causal diagram is partially corrupted.

Denoising Objective

One could model a conditional distribution over clean blocks. The reported implementation instead learns the deterministic reconstruction map

$$f_{\theta} : \tilde{B}(C, y) \mapsto \hat{B}(C, y).$$

Training minimizes a reconstruction objective of the form

$$\mathcal{L}_{\text{denoise}}(\theta) = \mathbb{E}_{B, \tilde{B}} \left[\|M \odot (B(C, y) - f_{\theta}(\tilde{B}(C, y), t))\|_2^2 \right], \quad (42)$$

where M marks the synthetically perturbed coordinates. The reported training code uses this masked mean-squared error; it does not add graph consistency, motif-preservation, or temporal-smoothness penalties.

This objective teaches inversion of the chosen synthetic corruption process. It neither forces a flat trajectory nor guarantees that the removed variation is an artifact of extraction rather than genuine change.

Block Denoising Encoder

Given a corrupted block

$$\tilde{B}(C, y) = (\tilde{X}(C, y), \dots, \tilde{X}(C, y + k)),$$

the model computes contextualized temporal representations

$$H = \text{Encoder}_{\theta}(\tilde{B}(C, y)).$$

The experiments reported here use a temporal Transformer with model width 192, four attention heads, three encoder layers, and blocks of three years. Other possible encoders include:

- a small temporal Transformer over yearly states,
- a graph-temporal encoder operating on yearly causal graphs,
- a KET-style temporal block encoder that aggregates across local time neighborhoods with structured kernels.

A prediction head then reconstructs the repaired block

$$\hat{B}(C, y) = g_\theta(H).$$

Sliding this block window across overlapping year ranges yields repaired states $\hat{X}(C, y)$ for the entire company trajectory.

Relation to Earlier Denoising Stages

This temporal denoiser generalizes the earlier block denoisers used in the DEMOCRITUS pipeline.

In Phase 1.5, the denoising problem acted on intra-document blocks of relational triples, repairing noisy local causal structure extracted from a single filing. Here, the objects are yearly causal states rather than individual triples:

$$(X(C, y), X(C, y + 1), \dots, X(C, y + k)).$$

Thus the two stages share the same principle of structured completion, but operate at different scales: Phase 1.5 repairs *intra-document* structure, while temporal diffusion repairs *inter-year* structure.

Categorical Interpretation

If **Year** is the path category on the observed years and the representation supplies a transition morphism for every generating year step, then a company defines a temporal functor

$$F_C : \mathbf{Year} \longrightarrow \mathbf{CausalState}, \quad F_C(y) = X(C, y).$$

The implementation supplies yearly vectors and estimates finite-dimensional transition maps. Calling the resulting assignment a functor records the intended typing; it does not prove that an independently specified family of temporal equations commutes. The denoiser acts as a numerical operator

$$\Phi : F_C \longmapsto \hat{F}_C,$$

mapping a noisy company functor to a repaired one whose local temporal diagrams are more coherent.

It may be interpreted as a candidate structured repair map, but its Transformer implementation is also a nonlinear smoother and must be compared with simpler smoothers and interpolants. Earlier chapters emphasized left-Kan-style aggregation as a language for combining evidence. Here one may use a right-Kan-like intuition: each repaired yearly state should be compatible with multiple local temporal constraints at once. No right Kan extension is computed in the reported code, however. This is an interpretive analogy, not an implementation theorem.

Output Artifacts

After inference, the implemented pipeline writes:

- repaired yearly causal states $\hat{X}(C, y)$,
- raw, denoised, and difference vectors for each company-year,
- top-coordinate summaries and raw-versus-denoised distances,
- block-level and run-level manifests.

These artifacts support later stages of the pipeline, including multi-company trajectory comparison, metric-space construction, and trajectory diagnostics such as the structural-indication radar.

Temporal Diffusion as a Schrödinger-Bridge-Like Repair Process

The temporal diffusion model introduced above also admits a useful probabilistic interpretation. At a high level, the denoising process may be viewed as a *Schrödinger-bridge-like* repair problem on trajectories of causal states. The point of this interpretation is not to claim that the present model solves a fully specified bridge problem in complete generality, but rather to clarify the variational principle that the repair process is trying to approximate: among all temporally plausible trajectories consistent with noisy local observations, prefer the one that departs as little as possible from a reference dynamics.

A Brief Reminder on Schrödinger Bridges

The classical Schrödinger bridge problem arises in stochastic control and entropy-regularized optimal transport. Given a reference stochastic process P on a state space $\mathcal{X}$, together with endpoint constraints such as prescribed initial and terminal marginals, one seeks a new process Q that matches the constraints while remaining as close as possible to the reference process. Formally, the bridge is the minimizer

$$Q^* = \arg \min_Q \text{KL}(Q \| P)$$

among all admissible processes Q satisfying the specified constraints.

The resulting process may be interpreted as the *least surprising* or *minimum-information* modification of the reference dynamics that is compatible with the observations. This viewpoint has become increasingly important in modern generative modeling, where diffusion and denoising procedures are often understood as approximating entropy-regularized path-space inference.

Temporal Blocks as Local Trajectory Fragments

In the present setting, the state space is the space of yearly causal states extracted from corporate filings. For a company C and year y , let

$$X(C, y)$$

denote the corresponding causal object.

A temporal block

$$B(C, y) = (X(C, y), X(C, y + 1), \dots, X(C, y + k))$$

may then be interpreted as a short trajectory fragment in this state space.

Because the yearly states are extracted from text and therefore noisy, the observed block should be treated as an imperfect measurement of an underlying latent temporal process. After corruption or extraction noise, we obtain a noisy block

$$\tilde{B}(C, y),$$

which acts as a partially observed local trajectory.

A Variational Interpretation of the Denoiser

The implemented denoiser produces a candidate reconstructed local trajectory

$$\hat{B}(C, y) = f_\theta(\tilde{B}(C, y)).$$

This reconstruction suggests a variational comparison. Suppose there is a reference temporal evolution P describing ordinary or expected year-to-year movement in corporate causal state space. An actual Schrödinger bridge would search for a repaired local trajectory distribution Q that is:

- consistent with the noisy observations encoded in $\tilde{B}(C, y)$, and
- as close as possible to the reference dynamics P .

At the level of intuition, this is the same principle as the Schrödinger bridge:

$$Q^* \approx \arg \min_Q \text{KL}(Q \| P) \quad \text{subject to compatibility with the observed corrupted block.}$$

The present model does not specify endpoint marginals, learn a reference path measure P , or minimize $\text{KL}(Q \| P)$. It minimizes masked reconstruction error under synthetic corruption. It is therefore *Schrödinger-bridge-like* only as a design motivation; minimum-information optimality has not been established.

Why This Interpretation Is Useful

This perspective explains several qualitative features of the repair process.

- **Temporal coherence.** Among multiple candidate repairs compatible with the noisy block, the bridge-like principle favors those that require the smallest deviation from ordinary temporal evolution.
- **Suppression of isolated artifacts.** Single-year spikes, missing motifs, or spurious edges are discouraged if they would force a large deviation from the reference dynamics.
- **Recovery of missing structure.** A genuine bridge model could restore a relation absent in one year when the surrounding years and observation model make it probable.
- **Comparability across companies.** A validated repair model could reduce extraction artifacts before downstream trajectory comparison.

These are desiderata for a bridge formulation, not consequences of the current reconstruction loss.

Categorical View: Repair of Temporal Functors

The same idea may be expressed in the categorical language used throughout the chapter. Each company trajectory defines a functor

$$F_C : \mathbf{Year} \longrightarrow \mathbf{CausalState},$$

while extraction noise or corruption yields a perturbed functor

$$\tilde{F}_C.$$

Temporal diffusion then acts as a repair operator

$$\Phi : \tilde{F}_C \longmapsto \hat{F}_C,$$

producing a repaired functor whose local temporal diagrams are more coherent than those of the noisy input. From this viewpoint, the Schrödinger-bridge intuition says that Φ should perform a *minimal-information repair*: modify the noisy temporal functor only as much as is necessary to restore local compatibility and temporal plausibility.

This also aligns well with the broader right-Kan intuition developed in earlier chapters. The repaired yearly state should not be viewed merely as an average of its neighbors, but as a state that is compatible with multiple local constraints at once. Temporal diffusion therefore combines a probabilistic bridge viewpoint with a categorical compatibility viewpoint.

Interpretive Consequences for Corporate Geometry

Under this reading, a repaired corporate trajectory is neither a raw text extraction nor a hand-smoothed time series. It is the output of a structured inference procedure balancing two forces: faithfulness to local observations and fidelity to plausible temporal evolution.

This has important consequences for the geometry of corporate trajectories:

- genuine strategic transitions remain visible,
- noisy one-year anomalies are less likely to dominate the geometry,
- distances between firms become less sensitive to extraction noise,
- trajectory-level diagnostics become more interpretable.

The Schrödinger-bridge viewpoint is accordingly a proposal for a stronger future model, not an empirical description of the current loss. Its value here is to state what additional path-space objects and constraints would be needed to turn the analogy into a theorem.

Multi-Company Metric Spaces over Causal Trajectories

Once repaired causal trajectories have been constructed for individual companies, the next natural step is to compare them across firms. This leads to a *multi-company metric space* whose points are not static feature vectors, but temporally repaired corporate causal trajectories.

The key idea is simple. A company is represented not by a single yearly snapshot, but by a time-indexed path through causal state space. To compare two firms, we therefore compare both their yearly states and the way those states evolve over time.

Company Trajectories as Functors

For each company C , temporal diffusion produces a repaired sequence of yearly causal states

$$\hat{X}(C, y), \quad y \in \mathbf{Year}.$$

As discussed earlier, this sequence is naturally viewed as a functor

$$\hat{F}_C : \mathbf{Year} \longrightarrow \mathbf{CausalState}, \quad \hat{F}_C(y) = \hat{X}(C, y).$$

Different companies therefore give rise to different functors from the year category into the category of causal states. Comparing companies becomes a problem of comparing these temporal functors.

Alignments Between Corporate Trajectories

To compare two firms C and D , we seek yearwise alignment maps

$$\eta_{C,D,y} : \hat{X}(C, y) \longrightarrow \hat{X}(D, y),$$

which express how the causal state of company C at year y may be matched to that of company D .

Taken together, these maps define a candidate comparison morphism

$$\eta_{C,D} : \hat{F}_C \Rightarrow \hat{F}_D.$$

In a category with specified transition morphisms, this family would be a natural transformation only if the following squares commute. That is, the alignments would commute with temporal evolution. For each year y , we would like the square

$$\begin{array}{ccc} \hat{X}(C, y) & \xrightarrow{T_{C,y}} & \hat{X}(C, y+1) \\ \downarrow \eta_{C,D,y} & & \downarrow \eta_{C,D,y+1} \\ \hat{X}(D, y) & \xrightarrow{T_{D,y}} & \hat{X}(D, y+1) \end{array}$$

to commute, where $T_{C,y}$ and $T_{D,y}$ denote the temporal transition operators for the two companies.

The prototype instead fits linear maps in a shared low-rank vector space and measures the residual *naturality defect*

$$\Delta_{C,D,y} = \eta_{C,D,y+1} \circ T_{C,y} - T_{D,y} \circ \eta_{C,D,y}.$$

A small residual is evidence only relative to the selected basis, fitted maps, normalization, and norm. It is called an approximate naturality defect, but it does not establish a natural transformation between semantic categories.

Distances Between Corporate Trajectories

Using the repaired trajectories and the alignment maps, we define three complementary notions of inter-company dissimilarity.

STATE DISTANCE. The first compares yearly repaired causal states directly:

$$d_X(C, D) = \frac{1}{|\mathbf{Year}|} \sum_y \text{dist}(\hat{X}(C, y), \hat{X}(D, y)).$$

Typical choices for dist include cosine distance between state embeddings, Wasserstein-type distances between edge-weight profiles, or divergences between motif distributions.

TRANSITION DISTANCE. The second compares year-to-year causal dynamics:

$$d_T(C, D) = \frac{1}{|\mathbf{Year}|} \sum_y \text{dist}(T_{C,y}, T_{D,y}).$$

This captures whether two firms change in similar ways, even if their absolute states differ.

NATURALITY DISTANCE. The third measures the extent to which a proposed alignment respects the dynamics:

$$d_\eta(C, D) = \frac{1}{|\mathbf{Year}|} \sum_y \|\Delta_{C,D,y}\|.$$

This is the most explicitly categorical term. It penalizes pairs of firms whose temporal evolution cannot be matched by a coherent structural correspondence.

A Composite Corporate-Trajectory Distance

These ingredients can be combined into a single trajectory-level dissimilarity

$$d(C, D) = \alpha d_X(C, D) + \beta d_T(C, D) + \gamma d_\eta(C, D),$$

where $\alpha, \beta, \gamma \geq 0$ control the relative importance of state similarity, dynamical similarity, and naturality.

Strictly speaking, this is a metric only when the component distances and alignment procedure satisfy the appropriate conditions. In practice, it is best viewed first as a *trajectory dissimilarity* or *pseudometric*, which can then be used to induce a geometric space of firms.

Such a dissimilarity could support clustering or retrieval after its metric properties and empirical stability are established. The reported panel does not yet validate those downstream uses.

Why Repair Comes First

The transformed trajectories $\hat{F}_C$ define the proposed comparison. If one compares raw yearly extractions directly, the resulting distances are easily dominated by extraction noise, reporting-style artifacts, or single-year anomalies. Temporal denoising is intended to suppress such artifacts before cross-company comparison, but the holdout results below show that the current models have not yet established that advantage.

Thus the proposed space is a geometry over transformed document-derived states. At present, proximity means similarity under the chosen document-derived representation. Separating corporate structure from extraction and reporting artifacts requires independent validation.

Interpretation

This trajectory space provides a principled language for studying corporate evolution.

- Firms that lie close together are similar under the selected document-derived coordinates and weights.
- Firms with small approximate naturality residuals have fitted transition maps that are compatible under the selected linear alignment.
- Sectoral or strategic families may appear as clusters or low-dimensional manifolds in the resulting geometry.
- Outlier firms may be identified as those whose trajectories are hard to align naturally with any peer group.

In this way, the metric space turns repaired causal trajectories into a comparative geometry of firms.

Empirical Role in the Chapter

The next section illustrates this construction on preliminary multi-company experiments. The experiment below therefore distinguishes descriptive coverage from validation. It shows that the pipeline produces comparable artifacts, then tests whether the learned temporal models outperform relevant reconstruction and forecasting baselines.

What the Structural-Indication Score Measures

The displayed radar is a deterministic heuristic, not a learned survival model. Within each company, raw features are centered by their median, divided by their interquartile range, and passed through a logistic map to the interval $[0, 100]$. Five resulting components—instability I , transition break T , margin pressure M , growth fragility G , and opacity O —are combined as

$$R = 0.28I + 0.18T + 0.22M + 0.22G + 0.10O.$$

The weights are design choices. No bankruptcy, return, earnings, volatility, or human risk labels were used to fit them in the reported panel. Because the robust scaling is computed from each company's own time series, a score of 70 primarily means "high relative to this company's history under these features." Cross-company rankings require additional calibration and should not be read as probabilities or investment advice.

Empirical Illustration: A Multi-Company Panel of Corporate Trajectories

To examine the temporal diffusion framework beyond individual case studies, we constructed repaired causal trajectories for a panel of **26 companies** using annual 10-K filings. This moves the chapter beyond the earlier two-company illustration and lets us study corporate geometry over a genuine cross-company population rather than over a single pair of trajectories.

For each company C and fiscal year y , the brand-aware DEMOCRITUS pipeline extracts a yearly causal atlas

$$X(C, y)$$

from the corresponding filing. These yearly states contain weighted causal relations, motif summaries, latent embeddings, and regime-level diagnostics. Temporal diffusion then repairs the raw sequence

$$X(C, y_0), \dots, X(C, y_T)$$

into a denoised sequence

$$\hat{X}(C, y_0), \dots, \hat{X}(C, y_T),$$

yielding for each firm a repaired temporal functor

$$\hat{F}_C : \mathbf{Year} \rightarrow \mathbf{CausalState}.$$

As of March 13, 2026, the observed panel contains 26 companies and 597 company-year observations spanning 2002–2026. A companion dense panel contains 650 rows on a common 2002–2026 grid, but the factual summaries below refer to the observed panel rather than the interpolated one. All 26 companies have temporal diffusion metrics and structural-indication radar scores, and 43 pairwise functor analyses are currently available. Coverage is complete for all 26 companies through 2025, while only 4 companies currently extend into 2026 (22 end in 2025). The latest snapshot is therefore asynchronous. The panel supports descriptive cross-sectional summaries, but unequal endpoints must be retained in every interpretation. These counts and the results below are documented in the Brand-Aware DEMOCRITUS study [Mahadevan, 2026a].

Middle-Year Reconstruction Benchmark

Training loss on synthetically corrupted blocks does not test whether a model can recover an unobserved company-year. The strongest available test therefore withholds the middle state of every three-year block, zeros its vector and associated context, reconstructs it from the two neighboring years, and scores the prediction against the observed filing-derived state. The observed state is itself an extraction artifact rather than causal ground truth, but the protocol at least prevents the model from copying the target.

Evaluation	Method	Targets	L2 ↓	Cosine ↑	Top-edge recall ↑
26-company middle-year holdout	company-local baseline	545	0.2194	0.5313	0.4220
26-company middle-year holdout	pooled temporal model	545	0.3004	0.0282	0.3540
Adobe middle-year holdout	temporal baseline	23	0.2183	0.5464	0.4203
Adobe middle-year holdout	KET-local variant	23	0.3199	0.0350	0.4094

Experiment: brand-aware temporal walk-forward audit

The risk-indication layer also has a walk-forward diagnostic, but its target is the next value of the same heuristic score, not an external event such as bankruptcy, earnings surprise, or realized volatility. On 202 observed-only predictions, an intrinsic-history baseline obtains mean absolute error 10.5595, whereas adding the peer-panel term yields 10.9197. The peer term therefore worsens this internal forecasting task by 0.3602 points. The experiment is useful as a falsification check: current evidence does not show that pooling across companies improves either held-out state reconstruction or next-score prediction.

Table 18: Leakage-resistant temporal-imputation results. Lower L2 and higher cosine and top-edge recall are better. The pooled model loses to the company-local baseline on L1, L2, and cosine for all 545 targets; the KET-local variant likewise loses to the Adobe baseline on those metrics for all 23 targets. These negative results supersede comparisons that merely measured how little an unmasked inference output changed its own input.

Panel-Wide Structure

The following summaries describe values produced by the fixed heuristic radar. They do not establish that the model has recovered corporate risk or excluded artifacts of reporting and extraction. The latest-year ranking in the table is led by Goldman Sachs (87.7 in 2025), Boeing (81.7 in 2026), IBM (78.3 in 2026), Walmart (75.0 in 2025), and Cisco (74.9 in 2025). The differences are large on the score's internal 0–100 scale, but that scale is constructed from robustly rescaled features and is not a calibrated probability.

Company	Latest Year	Latest Indication	Peak Year	Peak Indication
Goldman Sachs	2025	87.7	2025	87.7
Boeing	2026	81.7	2026	81.7
IBM	2026	78.3	2026	78.3
Walmart	2025	75.0	2025	75.0
Cisco	2025	74.9	2025	74.9
Home Depot	2025	68.5	2025	68.5
Visa	2025	67.6	2025	67.6
JPMorgan	2026	66.7	2026	66.7
Amgen	2025	66.2	2024	68.7
Coca-Cola	2025	62.5	2024	82.3

Table 19: Highest latest-year structural-indication readings in the 26-company observed panel. Latest years are asynchronous, so the table mixes 2025 and 2026 endpoints.

At the level of long-run averages, the ordering compresses substantially. The highest mean-indication firms are Amazon (52.2), American Express (52.1), Walmart (52.0), Chevron (51.9), and IBM (51.7). This gap between endpoint rankings and panel averages records the distinction between persistent and endpoint-heavy values of the constructed score. Interpreting those

Year	Companies	Mean Indication	Indication 60+
2021	26	50.1	1
2022	26	47.0	0
2023	26	47.5	2
2024	26	62.8	17
2025	26	58.0	12
2026	4	67.1	3

patterns as stress or regime change requires external corporate-event labels.

Cross-Sectional Regimes in Recent Years

The largest descriptive panel shift is temporal rather than purely cross-sectional. The above table summarizes the best-covered recent years. In 2022 the panel-wide mean structural-indication score is only 47.0, with no companies above the 60-point threshold; in 2023 the mean remains low at 47.5, with only two such firms. By contrast, 2024 shows a broad high-score regime: the mean rises to 62.8 and 17 of the 26 companies exceed the threshold. In 2025 the panel partially relaxes, but it remains elevated, with mean 58.0 and 12 firms still above 60.

The composition of these score profiles also changes. In 2024 the mean panel signal is loaded heavily on transition break (77.2), margin pressure (76.7), and opacity (65.0). This is consistent with one possible broad-stress interpretation but does not identify it. In 2025 transition break remains high at the panel mean level (83.9), and growth fragility remains elevated (62.7), but mean margin pressure falls to 38.1. Thus the hand-specified components distinguish different kinds of internal model change. Whether those distinctions track external risk remains open.

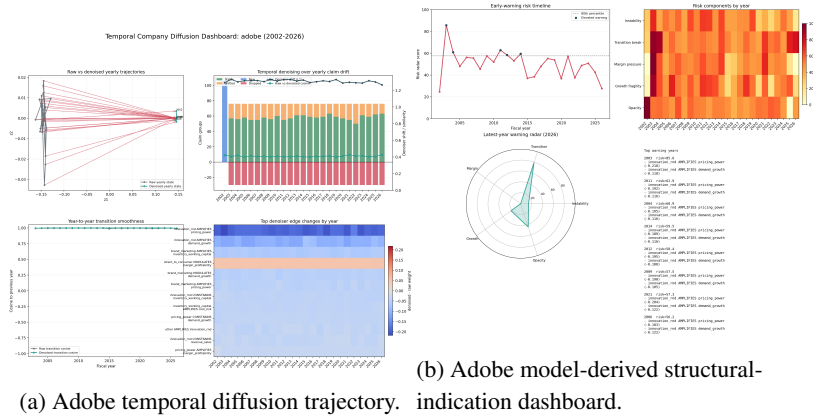
Temporal Diffusion and Multi-Company Geometry

Once transformed trajectories are available, they can be embedded in the multi-company trajectory space developed in the previous section. This produces a geometry over firms in which proximity is defined by the selected representation and dissimilarity. Whether this construction suppresses extraction noise while preserving genuine corporate change is a validation question.

At the panel level, several geometric questions become accessible:

- Do firms cluster by sector or business model?
- Which industries exhibit the most volatile causal trajectories?
- Which firms appear as geometric outliers because they undergo sharp regime changes?
- Does temporal diffusion improve cross-firm separation by reducing within-trajectory noise?

Table 20: Panel-wide cross-sectional summaries for the best-covered recent years. The column “Indication 60+” counts companies with structural-indication score at least 60; this is an internal threshold, not an externally calibrated risk category.



The reported experiments do not use sector labels, a retrieval benchmark, or independently defined strategic-shift labels to answer these questions. The pairwise comparison layer should still be interpreted cautiously. The panel now provides broad company coverage and many pairwise comparisons, but several heterogeneous pairs remain too similar in the present shared-basis representation, indicating that the cross-company ontology still needs a stronger support-aware refinement.

Case Studies: Adobe and Nike

Although the empirical core of the chapter is now the 26-company panel, it is still useful to retain named trajectories for interpretability. Adobe and Nike remain helpful because they occupy different parts of the panel distribution.

ADOBE. Adobe’s transformed trajectory has relatively small changes under the selected diagnostics. Its latest structural-indication score is 41.8 in 2026, below the largest endpoint values in the table. Its peak occurs much earlier, in 2003, at 68.9, and its low point is 33.3 in 2016. In the latest year the main signal is not margin pressure (which is essentially zero), but transition break (81.9) combined with moderate instability. This describes the internal components of the score; it is not an empirical finding about Adobe’s financial condition.

NIKE. Nike’s constructed score is more variable. Its latest value is 52.0 in 2025, but its peak rises to 74.5 in 2024. In the latest year Nike combines a very high transition-break score (88.1) with high growth fragility (81.4), even though margin pressure is again near zero. This makes Nike a descriptive contrast with Adobe within the model’s coordinate system, without implying an externally validated ordering of corporate risk.

Figure 36: Adobe as a representative transformed trajectory in the broader panel. The left plot compares raw and denoiser-produced yearly vectors. The right plot is the static dashboard generated from the same company-year indication table used by the Adobe structural-indication animation (whose preserved filename uses the earlier “risk radar” terminology); analogous animations were produced for Apple, IBM, NVIDIA, and Nike. These panels visualize model-derived quantities and are not observed survival outcomes or calibrated risk forecasts.

Summary of the Empirical Picture

The expanded panel demonstrates coverage rather than validated superiority. The present evidence supports four narrower claims:

- the pipeline produces typed company-year vectors and transformed trajectories for 26 firms and 597 observed company-years;
- the heuristic indication layer separates persistent from endpoint-heavy patterns on its own scale;
- the 2024 and 2025 score distributions differ descriptively from 2022 and 2023, without yet identifying an external economic regime;
- retrospective holdout tests do not show an advantage for pooled temporal denoising or the tested KET variant over their stated baselines.

In short, the empirical story is now

- extract yearly states → repair temporal blocks
- compare repaired trajectories
- derive trajectory-level diagnostics.

The result is a reproducible research representation and a set of falsifiable questions, not yet a validated geometry of corporate causality or risk.

Suggested Reading

For the probabilistic foundations of stochastic trajectories, see Billingsley [1995]. The Schrödinger problem and its connection to optimal transport are surveyed by Léonard [2014]. For causal influence along trajectories, Janzing et al. [2013] provides a useful complementary perspective. The temporal DEMOCRITUS prototype, panel construction, and descriptive artifacts are documented in Mahadevan [2026a]. The construction is developed further as part of the integrated PROMETHEUS research architecture [Mahadevan, 2026h]; ODYSSEY supplies the broader foundry semantics for preserving local provenance, verification, and truth conditions across those evolving artifacts [Mahadevan, 2026g].

Building Agentic Systems using Kan Extension Transformers

The previous chapter showed how yearly 10-K disclosures can be mapped into causal states and then repaired into temporal trajectories. In this chapter we describe a parallel construction at the level of *operational plans*. The goal is no longer to recover causal relations of the form “X influences Y,” but rather to infer short workflow fragments describing what the firm is trying to do: invest, innovate, optimize, manufacture, distribute, market, and ultimately realize revenue. This is the role of BASKET. Once these candidate workflows are extracted, a second system, ROCKET, reranks them against realized financial outcomes so that the retained plans are not only linguistically plausible, but also more tightly aligned with what the company actually went on to achieve.

Earlier chapters already introduced Kan Extension Transformers (KETs), the DEMOCRITUS machinery, and the 26-company panel. We therefore focus here only on the new empirical layer: extracting yearly workflow plans from 10-K filings, organizing them into a plan manifold with PLAN-KET, and then using ROCKET to select the plans that best match realized company performance.

Operational Plans as String Diagrams

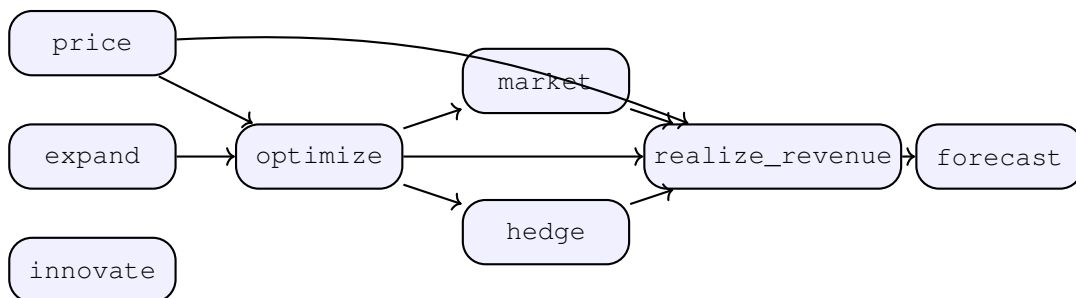
A recurring theme in this chapter is that operational plans are not best viewed as flat action sequences. Even when a workflow is written compactly as

`invest → optimize → expand → realize_revenue,`

this chain is usually only a convenient linearization of a richer underlying structure. In the raw 10-K text, several actions may be conditionally related, may proceed in parallel, or may jointly feed a later stage without being fully ordered among themselves. The correct mathematical object is therefore not a word or list, but a finite partially ordered plan.

Formally, we represent an operational plan by a finite set of action instances equipped with a precedence relation

$$\Pi = (A, \preceq),$$



where $a \preceq b$ means that action a must occur before action b in the plan. A displayed arrow chain is then only a *linear extension* of this partial order, not the plan itself. This distinction is important because many of the most interesting workflows extracted from 10-K filings contain branching, merging, and loosely coupled stages that disappear when they are flattened into a single chain.

This is exactly where string diagrams become useful. In the symmetric monoidal category $\mathcal{P}$ of plans, actions are morphisms, sequential composition is drawn horizontally, and independent or weakly coupled sub-operations are drawn on separate strands. A merge into a later action such as `realize_revenue` indicates that several upstream activities all contribute to a common downstream business outcome. In this way, the diagram makes the partial order visible rather than hiding it inside a linear notation.

The figure shows a representative example. Here `price` and `expand` both feed into `optimize`; from there the plan branches toward `market`, `hedge`, and directly toward `realize_revenue`; finally, `forecast` appears as a downstream continuation after realized business outcomes. The point of the figure is not that every corporate workflow has exactly this shape, but that aggregate plans are more faithfully represented as precedence-respecting compositional diagrams than as ordinary directed graphs.

This string-diagram view is not merely presentational. It is the representational form that BASKET attempts to recover from text, and it is the object that PLAN-KET and ROCKET subsequently embed, repair, and rerank.

BASKET: Extracting Operational Plans from 10-K Filings

BASKET treats each relevant 10-K paragraph as a local description of an operational workflow. A paragraph is mapped to a small compositional diagram of normalized actions, typically represented as a directed acyclic graph with an induced partial order. For example, language about capital expenditure, process improvement, channel expansion, and monetization can yield a workflow whose one-line linearization is

```
invest → optimize → expand → realize_revenue,
```

Figure 37: A precedence-respecting string-diagram style view of an aggregate operational plan. The diagram emphasizes partial order, branching, and merging structure rather than forcing the workflow into a single linear chain.

Quantity	Value
Yearly filing artifacts processed	599
Statement rows scanned	402,037
Extracted statement-level workflows	56,498
Action steps across all workflows	140,903
Average actions per workflow	2.49
PLAN-KET blocks	27,907
Normalized action vocabulary size	17
Discovered macro-skills	128
Training / validation blocks	22,325 / 5,582
Best validation loss	0.081

Table 21: BASKET corpus and PLAN-KET training summary for the raw-PDF full-panel run completed on March 18, 2026.

although the underlying extracted object is generally a small partially ordered diagram rather than a strictly linear chain.

These local workflows are then pooled within each company-year to form a yearly plan graph $P(C, y)$, and PLAN-KET embeds this graph into a latent plan space:

$$Z(C, y) = \Phi_{\text{PLAN-KET}}(P(C, y)).$$

Importantly, the extracted workflow need not be strictly linear. In many cases the underlying structure is better represented as

$$(\text{digitize} \otimes \text{optimize}) \rightarrow \text{expand} \rightarrow \text{realize\_revenue},$$

where $\otimes$ denotes independent or parallel sub-operations. The extractor represents such workflows by finite directed acyclic graphs whose incomparable actions can be displayed in parallel. This representation is compatible with a string-diagram interpretation, but the experiment does not establish the coherence axioms of a symmetric monoidal category. The categorical structure is therefore a model of the plan representation, not a structure inferred from language merely by observing parallel branches.

The latest run uses the raw-PDF extraction path, rather than the older relational-triple path. Thus BASKET now reads each annual filing directly, extracts section-aware paragraphs, converts them into normalized workflow candidates, mines recurring macro-skills, and forms short plan blocks for repair training. The resulting corpus is large enough that workflow geometry can now be studied on the same multi-company panel used earlier for causal trajectories.

The table records one reproducible partial-order extraction and training run. The above table shows that the extracted workflow corpus is already substantial. From 599 yearly filing artifacts, BASKET produced 56,498 statement-level workflows comprising 140,903 individual action tokens. These were organized into 27,907 short plan blocks for PLAN-KET repair training. The learned denoiser reached a best composite validation loss of 0.08098. This value is meaningful relative to the loss decomposition and corruption protocol used in that run. Without a matched baseline, it demonstrates

optimization on held-out blocks rather than, by itself, workflow accuracy or real-world executability.

The Learned Action Vocabulary

The normalized action vocabulary contains only 17 actions, but these are expressive enough to capture a wide range of corporate operational language. In the March 18, 2026 extraction run, the most frequent actions were `realize_revenue` (53,090 traces), `optimize` (40,272), `digitize` (8,192), `expand` (7,772), and `price` (5,419). This already reveals an important regularity: annual reports are dominated not by isolated descriptions of products, but by reusable action patterns that connect efficiency, expansion, commercialization, and financial realization.

The mined macro-skills make this even clearer. The most common recurrent subplans are

```
optimize → realize_revenue,
expand → realize_revenue,
realize_revenue → forecast,
```

followed by patterns such as

```
digitize → realize_revenue,
market → realize_revenue,
price → optimize.
```

Thus the extracted plan manifold is not arbitrary. It organizes around a small number of stable corporate motifs: optimization before monetization, expansion before monetization, and realized revenue feeding into forward guidance or capital-allocation decisions.

Company-Year Plan Graphs

Aggregating statement-level workflows within a year produces a company-year plan graph, which is the operational analogue of the yearly causal state from the previous chapter. These graphs are already interpretable at the firm level. For instance, Cisco's early filings are dominated by the short motif

```
optimize → realize_revenue,
```

whereas Chevron's recent filings show a richer industrial chain involving `invest`, `innovate`, `optimize`, `manufacture`, `expand`, `market`, and `realize_revenue`. In other words, BASKET does not merely count business verbs. It recovers different operational signatures across firms and across years.

This is the key extension beyond the earlier brand-aware DEMOCRITUS results. The causal pipeline asked what structural dependencies are present

in the filing. BASKET asks what the company is actually trying to do, expressed as a compositional workflow diagram. It therefore moves the representation of a firm from static causal discourse toward an explicitly agentic semantics.

Kan Extension Interpretation

The extraction and repair process admits a natural categorical formulation. Let $i : \mathcal{O} \hookrightarrow \mathcal{P}$ denote the inclusion of observed workflow fragments into the space of full plans, and let $F : \mathcal{O} \rightarrow \mathbf{Vect}$ assign embeddings to these fragments.

The key point is that the problem is not merely to “aggregate neighboring actions,” as in an ordinary graph network. The real question is:

how should a representation defined only on partially observed workflow fragments be extended to a larger executable plan?

Kan extensions provide a canonical answer once the categories, inclusion functor, semantic functor, and required limits or colimits have been specified.

There are two complementary directions in which the extension can proceed:

- the left Kan extension $\text{Lan}_i F$ extends by *assembling* larger plans from observed fragments that map into the target plan;
- the right Kan extension $\text{Ran}_i F$ extends by *completing* or constraining a partial plan so that it remains compatible with downstream requirements.

Formally, for a plan node or subdiagram $p \in \mathcal{P}$,

$$(\text{Lan}_i F)(p) = \text{colim}_{(i \downarrow p)} F$$

aggregates the observed fragments that can contribute to building p , while

$$(\text{Ran}_i F)(p) = \lim_{(p \downarrow i)} F$$

collects the compatibility constraints imposed by the ways in which p must fit into downstream observed fragments. Thus $(i \downarrow p)$ records all the ways known subplans can assemble into p , whereas $(p \downarrow i)$ records all the ways p must be completed so that later parts of the plan still make sense.

In workflow language, the distinction is intuitive. Left Kan asks:

given the observed prefix of a plan, what larger action pattern can be assembled from familiar fragments?

Right Kan asks:

given the downstream goal of the plan, what missing step is required so that the workflow can terminate in a valid way?

The first signal is expansive and constructive; the second is restrictive and goal-driven.

A simple corporate example makes this concrete. Suppose a filing implies the partial plan

`optimize → ? → realize_revenue.`

From previously observed workflows, the prefix `optimize` often extends to actions such as

`expand,      market,      distribute.`

This is the left-Kan signal: plausible continuations assembled from the prefix. But the suffix

`? → realize_revenue`

imposes a different constraint. In the corpus, actions that most often feed valid inputs into `realize_revenue` include

`expand,      market,      price.`

This is the right-Kan signal: actions compatible with the downstream goal. The overlap of the two channels suggests that `expand` and `market` are not just locally plausible but also globally compatible repairs. In this way, PLAN-KET treats workflow completion as the intersection of assembly evidence from the prefix and compatibility evidence from the goal.

This is also why the same architecture can patch both *missing steps* and *invalid orderings*. If a workflow fragment is too short, the left-Kan channel contributes likely intermediate structure; if the fragment is incompatible with its intended downstream action, the right-Kan channel pushes the representation toward repairs that satisfy the future boundary condition. What emerges is not only a denoised node embedding, but a repaired local plan that is more executable as a compositional diagram.

PLAN-KET is interpreted as a learned neural surrogate for these categorical operations. The implementation does not compute a categorical Kan extension or verify its universal property. At each plan node v , it builds a local embedding together with a predecessor-assembly channel and a successor-constraint channel:

$$z_v^{\text{Lan}} \approx (\text{Lan}_i F)(v), \quad z_v^{\text{Ran}} \approx (\text{Ran}_i F)(v),$$

and fuses them with the local node representation,

$$h_v = \text{Fuse}\left(z_v^{\text{local}}, z_v^{\text{Lan}}, z_v^{\text{Ran}}\right).$$

The left channel says what the current action should inherit from its predecessors; the right channel says what it must satisfy in order to support its successors. In this sense, PLAN-KET is not merely a graph encoder. It is a learnable approximation to left-Kan plan assembly and right-Kan plan completion on the category of corporate workflows.

ROCKET: Financially Grounded Plan Selection

The output of BASKET is intentionally redundant. A single paragraph can often support several closely related workflows, and many of these are linguistically plausible. The purpose of ROCKETS is to choose, from this local neighborhood of candidates, the plan that best matches realized financial outcomes. Formally, for a base plan p_0 extracted from company C in year y , ROCKETS searches a neighborhood of edited plans and selects

$$p^*(C, y) = \arg \max_{p \in \mathcal{N}(p_0)} R(p; C, y),$$

where the reward R is defined from observed company outcomes.

While this formulation suggests a local reranking procedure, the empirical results show that ROCKETS performs a more structured transformation. Rather than replacing plans arbitrarily, it systematically inserts missing intermediate stages and extends short fragments into multi-step workflows. In this sense, ROCKETS acts as a *plan completion and normalization operator*, mapping shallow fragments into canonical compositional pipelines.

The financial experiment uses a second, section-aware extraction containing 56,698 plans rather than the 56,498-plan partial-order corpus summarized earlier in this chapter. Their extraction settings and learned models should not be interchanged. In this financial run, the section-aware + KET reward used five targets:

revenue growth, operating margin, free-cash-flow margin, return on assets, debt-to-assets,

evaluated primarily at the next-year horizon. The latest run also interpolates the financial reward with PLAN-KET at weight $\lambda = 0.20$, so that local candidate selection remains outcome-driven but is regularized by the learned plan manifold. Because these targets require a matched outcome table, the financially supervised evaluation is performed on the 588 aligned company-years for which plan features and outcome data are both available.

The table shows that the latest ROCKETS run makes more edits than the earlier section-aware reranker, but it still does not rewrite the extracted plans wholesale. It changes 13,083 of 56,698 statement-level plans, and the winning edits are now dominated by `macro_completion`, `macro_merge`, `local_insert`, and `local_merge`. This means that KET does not merely amplify the previous scoring rule. It changes which repairs are favored, leaning more heavily toward completion and partial-order repair while reducing fallback to evidence-matching alone. The most frequent inserted actions are `expand`, `optimize`, `market`, and `realize_revenue`, which is exactly what one would expect if the objective favors fuller commercialization chains over very short extracted fragments. Because the reported gains use the same reward that selected each candidate, they are optimization-score gains by construction; they are not an independent forecast, causal-effect estimate, or out-of-sample validation of the edited plans.

Quantity	Value
Statement-level plans reranked	56,698
Aligned company-years with outcomes	588
Plans changed by ROCKET	13,083
Changed-plan rate	23.1%
Mean score gain over all plans	+0.0073
Mean score gain on changed plans	+0.0316
Most common winning source	macro_completion (4,059)
Second winning source	macro_merge (3,434)
Third winning source	local_insert (2,185)
Fourth winning source	local_merge (1,994)
Most inserted action	expand (6,133)
Second inserted action	optimize (4,193)
Third inserted action	market (3,040)
Fourth inserted action	realize_revenue (3,030)

Table 22: Summary of the March 19, 2026 section-aware + KET financial ROCKET reranking run.

Company	Sector	Ranking score	Label
3M	industrials	+2.589	more_consistent
Home Depot	consumer	+1.781	more_consistent
UnitedHealth	healthcare	+0.875	suggestive
Visa	financials	+0.827	suggestive
Boeing	industrials	-2.321	more_consistent
Amgen	healthcare	-1.508	more_consistent
Caterpillar	industrials	-1.431	more_consistent
Amazon	consumer	-0.979	suggestive

Table 23: Representative company-level structure/outcome scores from the March 19, 2026 section-aware + KET ROCKET reports. Positive scores indicate that more branched, less linearly constrained yearly plan aggregates tend to align with stronger next-year outcomes for that firm; negative scores indicate the reverse.

Outcome Alignment

The crucial question is no longer only whether reranked plans improve local financial reward, but whether the resulting *aggregate workflow geometry* aligns more consistently with realized outcomes across the multi-company panel. To test this, we regenerate company-level structure/outcome reports from the March 19, 2026 section-aware + KET run. These reports cover 24 of the 26 firms in the panel; Goldman Sachs and JPMorgan are excluded from the ranking tables because this comparison requires at least three next-year outcome metrics for each company-year.

The updated panel picture is more nuanced than the earlier non-KET section-aware run. The label mix shifts from 6/2/16 to 5/4/15 across *more_consistent*, *suggestive*, and *inconclusive* firms. Positive alignment strengthens for 3M, Home Depot, UnitedHealth, Visa, and Cisco, while negative alignment sharpens for Boeing, Caterpillar, Amgen, and Amazon. The sector summaries show the same redistribution: consumer drops from a mean Spearman alignment of +0.130 to +0.021, industrials flips from +0.177 to −0.124, financials move from −0.245 to nearly neutral at −0.019, and energy softens from −0.383 to −0.074. Archetype summaries also shift in an interpretable way: *healthcare_services* and *payment_network* each gain a positive case, whereas *biopharma* gains

a negative case and `brand_distribution_franchise` loses its earlier positive tilt.

The main conclusion is therefore not that KET uniformly strengthens the old story, but that it redistributes where partial-order branching appears to align with realized outcomes. Some previously strong cases soften toward inconclusive; several middling cases become suggestive; and a few negative alignments become much sharper. This is exactly the kind of behavior one would expect if KET is changing the geometry of the candidate set rather than merely rescaling the old financial reward.

Representative Edits

The qualitative edits made by the latest ROCKET run are also revealing. For Microsoft in 2017, a short extracted workflow

`optimize → forecast`

is upgraded to

`optimize → realize_revenue → forecast.`

For Nvidia in 2007,

`forecast → return_cash`

becomes

`optimize → realize_revenue → forecast → return_cash.`

Adobe in 2018 exhibits the same pattern:

`optimize → forecast`

becomes

`optimize → realize_revenue → forecast.`

These edits insert value-realization stages between operational steps and downstream financial actions. They are plausible normalizations under the declared action vocabulary, but the experiment does not independently establish that the edited workflow is the company’s true plan.

At the company level, Nike provides a useful illustration. In the latest visualizer output, Nike has 2,082 statement-level plans, of which 515 (24.7%) are changed by section-aware + KET financial ROCKET. The dominant Nike winning source families are `local_merge` (140), `macro_completion` (123), `local_insert` (112), and `macro_merge` (71). The most common inserted Nike actions are `optimize` (223), `expand` (220), `realize_revenue` (80), and `market` (49). Thus the updated reranker shifts Nike’s extracted plans toward fuller commercialization and execution chains rather than isolated operational verbs.

Aggregate Plans Before and After ROCKET

The aggregate-plan visualizers make the effect of ROCKET especially clear because they compare the company-wide plan geometry before and after reranking. At this level, one can see that ROCKET is not merely changing isolated statements. It alters the *aggregate operational backbone* of a firm.

In the refreshed section-aware + KET outputs, the main movement is usually in *incomparability* rather than in raw node growth. Most company-years keep the same number of surviving actions after reranking, but the selected plan changes how many of those actions remain mutually incomparable and therefore how branched the yearly workflow becomes. The aggregate effect is thus driven more by edge rewiring and partial-order repair than by uniform graph expansion.

Nike remains a useful within-company illustration. Across most Nike years, the aggregate graph stays at eight nodes after reranking, but incomparability is redistributed year by year: $0.643 \rightarrow 0.893$ in 2014, $0.821 \rightarrow 0.857$ in 2015, $0.714 \rightarrow 0.786$ in 2022, and $0.786 \rightarrow 0.821$ in 2024, while 2019 moves in the opposite direction from 0.821 to 0.607. The point is not that ROCKET always pushes toward more branching. Rather, it selects the branching pattern that best fits the firm's realized next-year outcome profile, which is why Nike remains globally inconclusive even though many individual years are revised.

Home Depot shows what a positive case looks like under the same mechanism. Again the yearly graphs usually keep eight nodes, but selected incomparability rises in key years such as 2015 ($0.667 \rightarrow 0.893$), 2020 ($0.679 \rightarrow 0.750$), 2021 ($0.714 \rightarrow 0.857$), and 2023 ($0.893 \rightarrow 0.929$). These shifts line up with one of the clearest positive structure/outcome scores in the panel (+1.781), indicating that broader partial-order structure tends to co-occur with stronger next-year performance for this firm.

Boeing provides the contrasting negative case. Its node counts also stay almost flat, yet the selected aggregates frequently become less incomparable in weak years: 2019 falls from 0.857 to 0.750, 2021 from 0.893 to 0.750, 2023 from 0.857 to 0.714, and 2024 from 0.679 to 0.500. This yields the strongest negative panel score (-2.321). Visa, by contrast, is suggestive on the positive side: years such as 2019 and 2024 move from $0.750 \rightarrow 1.000$ and $0.786 \rightarrow 0.964$ while maintaining strong realized fundamentals. The key lesson is that the refreshed aggregate comparisons are driven less by graph expansion than by how ROCKET+KET redistributes comparability versus parallelism inside a largely fixed action set.

These aggregate comparisons are important because they show that ROCKET's financial supervision does not merely improve local statement fit. In the refreshed panel outputs, the geometry of the company-year manifold changes mainly through ordering and branching structure, not through uniform action inflation. In categorical terms, ROCKET refines the ex-

tracted plan poset into a more coherent compositional object by selecting the partial-order shape whose geometry best aligns with realized outcomes.

Emergent canonical workflow. Across companies and years, ROCKET consistently transforms short workflow fragments into richer partially ordered plans organized around the macro-pattern

efficiency $\rightarrow$ growth $\rightarrow$ deployment $\rightarrow$ value.

A common displayed path through these plans is

optimize $\rightarrow$ expand $\rightarrow$ market $\rightarrow$ realize_revenue,

but the full aggregate object typically contains additional branches and joins. This pattern appears across multiple firms and sectors. It is a candidate regularity of the corpus and reranking objective; distinguishing a genuine cross-sector strategy pattern from an extraction or vocabulary artifact requires an independent annotation study.

From Workflow Extraction to Agentic Corporate Geometry

ROCKET AS NORMALIZATION. The transformation induced by ROCKET can be interpreted as a mapping

$$\mathcal{P} \rightarrow \mathcal{P}_{\text{normal}},$$

where $\mathcal{P}_{\text{normal}}$ is a subcategory of plans that satisfy value-completeness and compositional coherence. In this sense, ROCKET acts as a learned normalization functor that factors arbitrary workflow fragments into canonical economic pipelines.

STOCHASTIC INTERPRETATION. If plans are interpreted as stochastic morphisms, then each plan

$$P(C, y) : X(C, y) \rightarrow X(C, y + 1)$$

defines a probabilistic transition between operational states. In this view, ROCKET learns to select plans that maximize expected reward under observed financial outcomes, yielding a compositional stochastic model of corporate strategy.

Taken together, BASKET and ROCKET define an agentic analogue of the earlier causal geometry pipeline. BASKET extracts a large panel of company-year workflow graphs from raw 10-K filings and embeds them using PLAN-KET into a latent plan manifold. ROCKET then applies an

outcome-aware reward to rerank local workflow candidates, improving alignment with realized firm performance on most evaluated targets.

This gives a new representation of the firm. Earlier chapters emphasized what the firm *is connected to* causally. The present chapter emphasizes what the firm *does*: invest, innovate, optimize, manufacture, distribute, market, hedge, and realize revenue. In this sense, Kan Extension Transformers provide a common language for two complementary forms of corporate semantics. The causal chapters describe how firms are situated in a landscape of causal dependencies; BASKET and ROCKET describe how firms traverse a landscape of operational plans.

The broader significance is for agentic systems. An intelligent system often has to infer not merely facts or causal relations, but partially specified plans, repair them, compare them, and score them against downstream outcomes. That is precisely what happens here. PLAN-KET supplies the structured latent space; BASKET supplies the extracted candidate workflows; and ROCKET supplies an external reward that selects the plans most aligned with realized value. This is the sense in which these experiments instantiate the idea of *building agentic systems using Kan Extension Transformers*.

From Corporate Geometry to Topos Causal Models

This chapter and the previous one developed a geometric perspective on corporate evolution. Starting from yearly textual disclosures, the brand-aware DEMOCRITUS pipeline produced causal states

$$X(C, y),$$

which were then repaired by temporal diffusion into smoother and more coherent states

$$\hat{X}(C, y).$$

Taken together, these repaired states define a temporal functor

$$\hat{F}_C : \mathbf{Year} \longrightarrow \mathbf{CausalState},$$

representing the evolution of company C through a space of causal configurations.

This perspective yields a new vocabulary for corporate analysis. A firm is no longer treated merely as a sequence of filings or a static collection of financial indicators. Instead, it is modeled as a path through causal state space. That geometric viewpoint supports several new tools:

- temporal diffusion for repairing noisy yearly causal snapshots,
- multi-company trajectory spaces for comparing firms structurally,
- trajectory-level diagnostics such as radar indicators of instability or stress.

The expanded multi-company experiments in this chapter suggest that this framework is already useful empirically. Temporal diffusion improves local coherence of corporate trajectories, and repaired trajectories support more stable comparisons across firms. In this sense, the chapter has shown how causal extraction and structured denoising can be combined to produce an interpretable geometry of corporate evolution.

The Limitation of a Single Global Causal State Space

At the same time, the present framework still relies on a simplifying assumption: that all yearly causal states can be represented inside a single global state space **CausalState**.

This assumption is useful for building trajectory-level geometry, but it is also restrictive. Real causal reasoning is deeply contextual. Different companies, domains, time periods, and document collections may involve different causal vocabularies, different relevant variables, and different local organizational principles. Forcing all such states into one global coordinate system risks erasing precisely the contextual distinctions that make causal reasoning meaningful.

The limitation becomes even more pronounced when one wants to integrate causal knowledge across heterogeneous sources: scientific papers, corporate filings, policy reports, biomedical studies, or legal documents. In such cases, there may be no single globally adequate causal state space at all.

From Global Geometry to Contextual Causal Semantics

This limitation motivates the next step in the book: the move from global causal geometry to *Topos Causal Models*.

A topos provides a language not for one global universe of objects, but for a family of related local contexts together with the maps that connect them. From this perspective, causal knowledge need not be forced into a single global graph or embedding space. Instead, one may work with collections of local causal models, each adapted to its own context, and then study how these local models fit together.

This is the natural next generalization of the ideas in the present chapter. Corporate trajectories may be viewed as local sections of a richer contextual structure. Temporal diffusion repaired trajectories within a fixed temporal domain; *Topos Causal Models* will ask how such local causal constructions can be organized, compared, and composed across heterogeneous contexts more generally.

Conceptual Continuity

There is a direct conceptual continuity between the two chapters.

- In the present chapter, noisy yearly causal snapshots were repaired into coherent temporal trajectories.
- In the next chapter, families of local causal models will be organized into coherent contextual systems.

Both chapters are therefore instances of the same broader theme of the book:

intelligence requires extending partial, local, and noisy structures into larger coherent systems.

In the corporate setting, this produced repaired trajectories and multi-company geometry. In the topos setting, it will produce a framework for contextual causal reasoning that does not assume a single global causal universe from the start.

Outlook

The combination of causal extraction, temporal diffusion, and categorical reasoning already provides a powerful framework for studying complex evolving systems. Corporate dynamics offer one compelling example, but the same pattern applies much more broadly: scientific discovery, technological change, institutional evolution, and policy dynamics all involve partial local causal knowledge that must be organized into coherent larger structures.

The next chapter develops this broader perspective explicitly. Topos Causal Models provide a principled mathematical setting in which causal knowledge can remain local, contextual, and heterogeneous, while still supporting comparison, composition, and inference. They therefore generalize the geometric constructions of the present chapter into a richer theory of contextual causal semantics.

Suggested Reading

Kan extensions and their universal properties are developed in [Mac Lane \[1971\]](#), [Riehl \[2017\]](#), and Kan's original paper [[Kan, 1958](#)]. Predictive-state representations provide the state-estimation background for workflow monitoring [[Littman et al., 2001](#), [Singh et al., 2004](#)]. For decision making under distributed information, see [Witsenhausen \[1975\]](#). The PROMETHEUS architecture situates 10-K analysis within an integrated causal-research workflow over documents, data, and models [[Mahadevan, 2026h](#)]. ODYSSEY develops the encompassing foundry architecture for maintaining these financial artifacts with local provenance, verification, and truth-preserving promotion rules [[Mahadevan, 2026g](#)].

Topos Causal Models

In this chapter, we place causal presentations in a presheaf or sheaf *topos*, where local-to-global gluing, subobjects, and an internal language are available.²² This enriches the representational language used for structural causal models (SCMs), but it does not imply that every category of SCMs is automatically a topos or that every subobject is a causal intervention. Those conclusions require explicit closure and intervention-admissibility assumptions. The precise universal result is the density theorem: a functor from a small causal category into a cocomplete semantic category has a colimit-preserving extension along the Yoneda embedding, unique up to natural isomorphism among such extensions. We also introduce the Mitchell–Bénabou language and Kripke–Joyal semantics available inside the chosen topos.

²² Sridhar Mahadevan. Universal causal inference in a topos. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025g. URL <https://openreview.net/forum?id=TOhpnECT10>

From Structured Completion to Topos Causal Models

The experiments of the previous chapters on structured language modeling and temporal diffusion over causal trajectories revealed a striking empirical phenomenon. Across multiple datasets and model architectures, language modeling becomes dramatically easier when the model is asked to *complete partially specified structures* rather than generate complete structures from scratch.

In direct block prediction the model must learn a map

$$C_t \rightarrow \Sigma^B$$

which corresponds categorically to a morphism into a product object. The model must choose an element of this product in a single step, which becomes increasingly difficult as the block size and vocabulary grow.

Denoising models instead learn maps of the form

$$(C_t, \tilde{x}) \rightarrow x,$$

where $\tilde{x}$ is a corrupted or partially specified block. The model therefore performs *structured completion*: extending a partial object into a coherent full object.

The DEMOCRITUS architecture described in the previous chapters operates under an analogous principle. Rather than constructing a global causal

graph directly from documents, the system aggregates local causal fragments into a partially specified graph and repeatedly refines it. Formally, the system applies a completion operator

$$F : (G_t, E) \rightarrow G_{t+1}$$

that transforms a partially specified causal structure G_t using evidence E extracted from documents.

The preceding chapter extended this idea to the domain of corporate analysis. There we interpreted companies as tracing trajectories through a high-dimensional space of causal states extracted from financial disclosures. Temporal diffusion models were used to repair noisy yearly snapshots and produce coherent trajectories

$$F_C : \text{Year} \rightarrow \text{CausalState}.$$

These repaired trajectories enabled the construction of multi-company metric spaces and trajectory diagnostics such as the survival radar introduced in the previous chapter.

Viewed geometrically, both structured language modeling and causal discovery can be interpreted as instances of *simplicial completion*. A corrupted token block corresponds to a partially specified simplex whose missing faces must be filled. Likewise, a partial causal graph contains incomplete simplices of causal relations that must be extended to produce a coherent global structure.

This perspective naturally leads to the notion of *horn completion* familiar from simplicial homotopy theory. In a Kan complex, every horn can be extended to a full simplex. Denoising language models and causal completion operators can be interpreted as learned mechanisms for performing analogous completion operations on structured data.

The categorical viewpoint developed throughout this book suggests that many intelligent systems operate by extending partial structures within a suitable category of information. Language modeling, causal discovery, and knowledge integration can all be understood as instances of this more general process.

This chapter develops this idea further through the framework of *Topos Causal Models*. Topos theory provides a natural mathematical setting for reasoning about collections of local structures and their global consistency. Within this framework, causal models can be viewed as objects in a topos whose internal logic captures the relationships among variables, interventions, and observational evidence.

In this sense, the structured completion procedures described in this chapter can be viewed as computational approximations to the gluing and consistency operations that arise naturally in topos-theoretic formulations of causal reasoning.

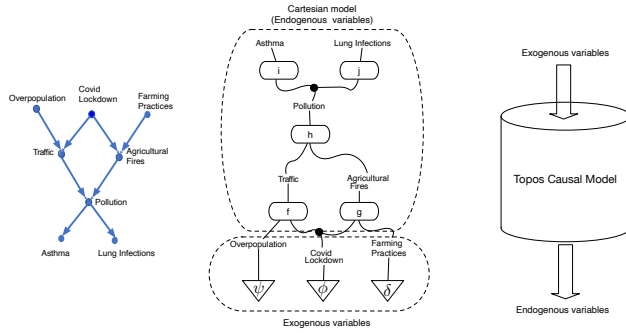


Figure 38: Topos causal models (TCMs) are defined as a category $\mathcal{C}_{TCM}$ whose objects $c \in \mathcal{C}_{TCM}$ are causal models, and whose arrows $\mathcal{C}_{TCM}(c, c')$ are commutative diagrams between models c and c' . A specific object c defining a model can be conceptualized as a DAG (left, where information flows from top to bottom), or a string diagram in a Markov category (middle, where information flows from bottom to top), or in terms of its induced unique “blackbox” function mapping exogenous variables to endogenous variables (right).

Introduction

In recent years, there has been significant interest in categorical models of causality. Categorical approaches fundamentally differ from past work in causality in their focus on the elucidation of universal properties. In our previous work, we introduced the framework of *universal causality* based on the notion of universal properties in category theory: a causal property is universal if it can be defined in terms of an *initial* or *final* object in a category of causal diagrams, or in terms of a *causal representable functor* using the Yoneda Lemma.²³ For example, a structural causal model (SCM) is defined as a (deterministic) mapping from a collection of exogenous variables into a collection of endogenous variables, derived by “collating” local functions that serve as independent causal mechanisms. However, SCMs can be further analyzed in terms of their universal properties, such as categorical product, coproduct, limits and colimits, equalizers and coequalizers etc. These latter properties can be shown formally to be initial or final objects in a category of diagrams, or as representable functors through the Yoneda Lemma.

Our main contribution in this chapter is a *topos-theoretic* language for causal models. A topos supports sheaf gluing, a subobject classifier, and an internal logical language. These structures can represent local mechanisms, context-sensitive predicates, and admissible intervention subobjects once the site and causal semantics have been specified. They enrich the usual SCM language; they do not replace the statistical assumptions required for causal identification.

The objects in a TCM category can be conceptualized in multiple ways. First, each object can be a causal graphical model. Each object can also be a functor: for example, directed graphs form a topos functor category. TCMs can also be defined in terms of *string diagrams* in a symmetric monoidal *Markov category*, where we restrict ourselves to the Markov subcategory defined through deterministic morphisms. For example, the arrow $h : \text{Traffic} \otimes \text{Agricultural Fires} \rightarrow \text{Pollution}$ defines a deterministic mapping specifying the two potential causes of Pollution. For exogenous variables, the arrow $\psi : I \rightarrow \text{Overpopulation}$ defines the marginal distribution on Over-

²³ Sridhar Mahadevan. Universal causality. *Entropy*, 25(4):574, 2023. DOI: 10.3390/E25040574. URL <https://doi.org/10.3390/e25040574>

population, where I is the terminal object in the Markov category. Finally, we can view a TCM object as a “blackbox” function that maps some collection of exogenous variables (e.g., “Overpopulation”, or “Farming Practices” into some set of endogenous variables, e.g., “Asthma” or “Pollution”).

Principles of Universal Causality

We give a brief overview of the fundamentals of universal causality (UC) before delving into the specific details of the TCM framework. UC rests on the Yoneda Lemma – any object in a category can be defined by the interactions it makes with other objects (upto isomorphism). In the setting of causal inference, it means that objects in a TCM category can be ascribed “meaning” through studying the arrows of the category, without having to “look inside” the object. The Yoneda Lemma states that the set of all morphisms into an object d in a category $\mathcal{C}$, sometimes denoted as $\mathbf{Hom}_{\mathcal{C}}(-, d)$, or as $\mathcal{C}(-, d)$, denoted as the *presheaf*, is sufficient to define d up to isomorphism. The category of all presheaves forms a *category of functors*, and is denoted $\hat{\mathcal{C}} = \mathbf{Set}^{\mathcal{C}^{op}}$. This category forms a topos, and will be fundamental to the TCM framework.

Lemma 2. Yoneda lemma: *For any functor $F : \mathcal{C} \rightarrow \mathbf{Set}$, whose domain category $\mathcal{C}$ is “locally small” (meaning that the collection of morphisms between each pair of objects forms a set), and any object c in $\mathcal{C}$, there is a bijection $\mathbf{Hom}(\mathcal{C}(-, c), F) \simeq Fc$ that associates a natural transformation $\alpha : \mathcal{C}(-, c) \Rightarrow F$ to the element $\alpha_c(1_c) \in Fc$. This correspondence is natural in both c and F .*

Definition 47. *A universal property of an object $c \in \mathcal{C}$ in a category $\mathcal{C}$ is expressed by a representable functor F together with a universal element $x \in Fc$ that defines a natural isomorphism $\mathcal{C}(-, c) \simeq F$. The collection of morphisms $\mathcal{C}(-, c)$ into an object c is called the **presheaf**, and from the Yoneda Lemma, forms a universal representation of the object.*

We state two key results that underly UC. While both these results follow directly from basic theorems in category theory, their significance for causal inference is what makes them particularly noteworthy. The first result pertains to the notion of diagrams as functors, and shows that for the functor category of presheaves, which is a universal representation of causal inference, every presheaf object can be represented as a colimit of representables through the Yoneda Lemma. This result can be seen as a generalization of the very simple result in set theory that each set is a union of one element sets. The second result is the causal reproducing property, which shows that the set of all causal effects between two objects is computable from the presheaf functor objects defined by them. Both these results are abstract, and apply to any category representation of a causal model.

Theorem 6. Universality of Diagrams in UC: *In the functor category of presheaves $\mathbf{Set}^{\mathcal{C}^{op}}$, every functor object F is the colimit of a diagram of representable objects, in a canonical way.*

To explain the significance of this result for causal inference, note that UC represents causal diagrams as functors from an indexing category of diagrams to an actual causal model. The theorem above tells us that every presheaf object can be represented as a colimit of (simple) representable objects, namely functor objects of the form $\mathbf{Hom}_{\mathcal{C}}(-, c)$.

Theorem 7. Causal Reproducing Property: *All causal influences between any two objects c and d can be derived from its presheaf functor objects, namely*

$$\mathbf{Hom}_{\mathcal{C}}(c, d) \simeq \mathbf{Nat}(\mathbf{Hom}_{\mathcal{C}}(-, c), \mathbf{Hom}_{\mathcal{C}}(-, d))$$

Any causal influence of an object c upon any other object d can be represented as a natural transformation (a morphism) between two functor objects in the presheaf category $\hat{\mathcal{C}}$.

Topos Causal Models

Definition 48. *The category $\mathcal{C}_{\mathcal{T}\mathcal{C}\mathcal{M}}$ of topos causal models is defined as a collection of objects $c \in \mathcal{C}_{\mathcal{T}\mathcal{C}\mathcal{M}}$, each of which is a triple $\langle U, V, F \rangle$ where $V = \{V_1, \dots, V_n\}$ is a set of endogenous variables, U is a set of exogenous variables, and F is a function from U to V . The arrows $\mathcal{C}_{\mathcal{T}\mathcal{C}\mathcal{M}}(c, d)$ are defined through commutative diagrams as illustrated below, where f and f' are the global functions induced by the TCM objects c and d , respectively.*

$$\begin{array}{ccc} U & \xrightarrow{h} & U' \\ \downarrow f & & \downarrow f' \\ V & \xrightarrow{g} & V' \end{array}$$

A submodel $c' = \langle U', V', F' \rangle$ of c is any subobject of c . The effect of an intervention on c is given by some submodel c' . Finally, let Y be a variable in V , and let X be a subset of V . The potential outcome in response to an intervention on X modeled by a submodel $c' \hookrightarrow c$ is the solution of Y in the submodel c' .

A commutative diagram, as the term suggests, is a structure showing the equivalence of two paths. Here, the diagram asserts that $g \circ f = f' \circ h$. In the context of our category $\mathcal{C}_{\mathcal{SC}\mathcal{M}}$, the arrow $f : U \rightarrow V$ is simply an SCM M , and f is its induced mapping from exogenous to endogenous variables. Similarly, f' is also the induced function mapping exogenous to endogenous variables for another SCM M' . The morphisms h and g are functions on SCMs, which transform one causal model into another. In the

specific case we are interested in, these functions define causal interventions, but in general, they may be arbitrary functions.

For completeness, we define a category $\mathcal{C}_{SCM}$ whose objects are indeed SCMs.

Definition 49. *The category $\mathcal{C}_{SCM}$ of structural causal models is defined as a collection of objects, each of which is a triple $\langle U, V, F \rangle$ where $V = \{V_1, \dots, V_n\}$ is a set of endogenous variables, U is a set of exogenous variables, F is a set $\{f_1, \dots, f_n\}$ of “local functions” $f_i : U \cup (V \setminus V_i) \rightarrow V_i$ whose composition induces a unique function F from U to V . Let X be a subset of variables in V , and x be a particular realization of X . A submodel $M_x = \langle U, V, F_x \rangle$ of M is the causal model $M_x = \langle U, V, F_x \rangle$, where $F_x = \{f_i : V_i \notin X\} \cup \{X = x\}$. The effect of an action $do(X = x)$ on M is given by the submodel M_x . Finally, let Y be a variable in V , and let X be a subset of V . The potential outcome of Y in response to an action $do(X = x)$, denoted $Y_x(u)$, is the solution of Y for the set of equations F_x .*

The set of arrows or morphisms between two objects c and d in the category $\mathcal{C}_{SCM}$, denoted $\mathcal{C}_{SCM}(c, d)$, represent ways of transitioning from SCM object c to d . For example, if d is a submodel of c , then the arrow defines a **do** calculus causal intervention.

Causal Models and the Arrow Topos

There is an important distinction between representing a causal model by its induced function and proving that a chosen category of causal models is a topos. Let $[1]$ be the walking-arrow category. Its objects are two vertices and one nonidentity arrow between them. A functor $[1] \rightarrow \mathbf{Set}$ is exactly a function $f : U \rightarrow V$, and a natural transformation is exactly a commutative square.

Theorem 8 (Arrow-topos theorem). *The category of set-valued arrows and commutative squares is the functor category $\mathbf{Set}^{[1]}$, and hence is a Grothendieck topos.*

Proof. Because $[1]$ is small, the functor category $\mathbf{Set}^{[1]}$ is a presheaf category (equivalently, $\mathbf{Set}^{(([1])^{op})^{op}}$). Every presheaf category on a small category is a Grothendieck topos. Limits and colimits are computed pointwise; exponentials and the subobject classifier are those of the functor topos. $\square$

This theorem justifies a topos of *functional presentations*. It does not automatically imply that a constrained category of structural causal models is a topos. Structural equations, acyclicity or unique-solvability conditions, distinguished variable decompositions, and the requirement that morphisms preserve causal mechanisms define additional structure. A subcategory cut out by those conditions is a topos only after closure under finite limits and exponentials and the existence of a subobject classifier have been established.

Nor is every subobject of a functional presentation a Pearl intervention. A hard intervention replaces selected structural mechanisms while leaving the others fixed. Treating an intervention as a monomorphism therefore requires an explicit admissibility condition encoding that mechanism-replacement semantics. The subobject classifier records subobjects in the ambient topos; it does not by itself certify causal identification or convert an arbitrary restriction into a do-operation.

Finally, intuitionistic logic does not arise from assigning an ad hoc third truth value to a commutative square. It arises from the Heyting algebra of subobjects in a topos. In a presheaf topos, truth values are sieves and may encode stage-dependent information; their structure is determined by the base category. This is the construction used in the remainder of the chapter.

Causal Models Over a Topos of Sheaves

We now describe a more general categorical framework for defining causal models as a topos by using the property that Yoneda embeddings of presheaves forms a topos. To ensure consistent extension into a unique global function, we build on the theory of sheaves, which ensures local functions can be “collated” together to yield a unique global function. In our setting, we will construct sheaves from categories over causal models through the Yoneda embedding $\mathcal{Y}(x) : \mathcal{C} \rightarrow \mathbf{Sets}^{C^{op}}$ and impose a Grothendieck topology.

Grothendieck Topology on Sites

Definition 50. A sieve for any object x in any (small) category $\mathcal{C}$ is a subobject of its Yoneda embedding $\mathcal{Y}(x) = \mathcal{C}(-, x)$. If S is a sieve on x , and $h : y \rightarrow x$ is any arrow in category $\mathcal{C}$, then

$$h^*(S) = \{g \mid \text{cod}(g) = D, hg \in S\}$$

Definition 51. A Grothendieck topology on a category $\mathcal{C}$ is a function J which assigns to each object x of $\mathcal{C}$ a collection $J(x)$ of sieves on x such that

1. the maximum sieve $t_x = \{f \mid \text{cod}(f) = x\}$ is in $J(x)$.
2. If $S \in J(x)$ then $h^*(S) \in J(y)$ for any arrow $h : y \rightarrow x$.
3. If $S \in J(x)$ and R is any sieve on x , such that $h^*(R) \in J(y)$ for all $h : y \rightarrow x$, then $R \in J(x)$.

We can now define categories with a given Grothendieck topology as *sites*.

Definition 52. A site is defined as a pair $(\mathcal{C}, J)$ consisting of a small category $\mathcal{C}$ and a Grothendieck topology J on $\mathcal{C}$.

Definition 53. The subobject classifier Ω is defined on any topos $\mathbf{Sets}^{C^{op}}$ as subobjects of the representable functors:

$$\Omega(x) = \{S \mid S \text{ is a subobject of } \mathcal{C}(-, x)\}$$

and the morphism $\mathbf{true} : 1 \rightarrow \Omega$ is $\mathbf{true}(x) = x$ for any representable x .

Universal Property of TCM over Functor Categories

Causal models, like SCMs, must represent both decomposable structure and (probabilistic) semantics. To capture this richer structure, we define TCM over *functor categories*, where every object is a functor that maps structure to semantics. For example, the category of Bayesian networks can be modeled as a functor category from a Markov category to the category **FinStoch** of finite stochastic processes.

Theorem 9. *Given a causal functor $A : \mathcal{C} \rightarrow \mathcal{E}$, such as the Bayesian network functor F_{CDU} , from a small category $\mathcal{C}$ (e.g., a symmetric monoidal category such as a Markov category) to a cocomplete category $\mathcal{E}$ (e.g., the category **Prob** of probability spaces (see Theorem 11)), the functor R from $\mathcal{E}$ to presheaves, given by (where $c \in \mathcal{C}$ and $E \in \mathcal{E}$)*

$$R(E) : c \mapsto \mathbf{Hom}_{\mathcal{E}}(A(c), E)$$

has a left adjoint $L : \mathbf{Sets}^{\mathcal{C}^{op}} \rightarrow \mathcal{E}$ defined for each presheaf P in $\mathcal{C}^{op}$ as the colimit

$$L(P) = \text{Colim} \left(\int_{\mathcal{C}} P \xrightarrow{\pi_P} \mathcal{C} \xrightarrow{A} \mathcal{E} \right)$$

where $\int_{\mathcal{C}} P$ is the category of elements, whose objects are pairs (c, p) , where c is an object of $\mathcal{C}$ and p is an element of $P(c)$ (recall P is a presheaf, i.e., a set-valued functor that maps each element c into a set), and its arrows are $(c', p') \rightarrow (c, p)$ for any morphism $f : c' \rightarrow c$ such that $pc = p'$.

Proof: Essentially, the theorem is stating that there is a pair of adjoint functors $L \vdash R$, defined as:

$$L : \mathbf{Sets}^{\mathcal{C}^{op}} \xrightarrow{\quad} \mathcal{E} : R$$

As defined earlier, a natural transformation between two functors $\tau : P \rightarrow R(E)$ is a family $\{\tau_c\}$ of maps indexed by the objects $c \in \mathcal{C}$, where each map τ_c is defined as the mapping:

$$\tau_c : P(c) \mapsto \mathbf{Hom}_{\mathcal{E}}(A(c), E)$$

which is natural in c . τ can also be defined as a set of arrows of $\mathcal{E}$ as $\{\tau_c(p) : A(c) \rightarrow E\}_{(c,p)}$ that is indexed by the objects (c, p) of the category $\int_{\mathcal{C}} P$ of elements of P . This fact implies that there is a bijection

$$\text{Nat}(P, R(E)) \simeq \mathbf{Hom}_{\mathcal{E}}(LP, E)$$

This bijection being natural in P and in E proves that L is a left adjoint functor to R . $\square$

Now, let us define a general causal functor as mapping from a decomposable symmetric monoidal category (e.g., a Markov category) to the symmetric monoidal category of probability spaces.

Definition 54. A causal functor $F : \mathcal{C} \rightarrow \mathbf{Prob}$ maps from a general symmetric monoidal category $\mathcal{C}$ with a comonoidal “copy-delete” structure to the category of probability spaces $\mathbf{Prob}$, where each object $(\Omega, \mathcal{F}, \mathbb{P})$ is a probability space, and the arrows are measure-preserving maps, namely $\mathbf{Prob}(c, d)$, where $c = (\Omega_c, \mathcal{F}_c, \mathbb{P}_c)$ and $d = (\Omega_d, \mathcal{F}_d, \mathbb{P}_d)$, where $f \in \mathbf{Prob}(c, d)$ is such that $\mathbb{P}_c(f^{-1}(A)) = \mathbb{P}_d(A)$ for all $A \in \mathcal{F}_d$.

Theorem 10. For each functor $A : \mathcal{C} \rightarrow \mathcal{E}$ from a small category $\mathcal{C}$ to a cocomplete category $\mathcal{E}$, the left Kan extension $L = \text{Lan}_{\mathcal{C}} A$ exists, preserves small colimits, and is equipped with a natural isomorphism $A \cong L \circ \mathcal{C}$. Moreover, L is unique up to unique natural isomorphism among colimit-preserving extensions with this property.

Proof. This is the standard free-cocompletion (density) property of the presheaf category. Pointwise, L is computed by the category-of-elements colimit in the preceding theorem. The Yoneda lemma gives the displayed natural isomorphism on representables, and preservation of colimits determines the extension on every presheaf because every presheaf is canonically a colimit of representables. $\square$

Theorem 11. For the particular category $\mathbf{Prob}$ of probability spaces and measure-preserving maps used in Definition 54, the required nonempty colimits exist under the construction cited below.

Proof. The coproduct and coequalizer constructions are given by van Belle [2024]. The restriction to nonempty diagrams matters: the definition used here has no initial probability space. Consequently, any application of Theorem 10 to this exact category must either use only the corresponding nonempty category-of-elements diagrams or pass to a specified cocompletion. The general theorem itself requires a genuinely cocomplete semantic category. $\square$

We can finally state the two central results of our chapter, the first (Theorem 12) establishes the universal property underlying TCMs, and the second (Theorem 13) shows that causal interventions define a Heyting algebra whose logic is intuitionistic.

Theorem 12. Any causal functor $F : \mathcal{C} \rightarrow \mathcal{E}$ from a small structural category $\mathcal{C}$ to a cocomplete semantic category $\mathcal{E}$ extends along the Yoneda embedding to $\text{Lan}_{\mathcal{C}} F$. The factorization is unique up to natural isomorphism among colimit-preserving extensions.

Proof. Apply Theorem 10. The adjective “causal” supplies the intended interpretation of F ; the mathematical conclusion follows from smallness and cocompleteness. $\square$

Definition 55. A Heyting algebra is a poset with all finite products and coproducts, which is Cartesian closed. That is, a Heyting algebra is a lattice, including bottom and top elements, denoted by $\mathbf{0}$ and $\mathbf{1}$, respectively,

which associates to each pair of elements x and y an exponential y^x . The exponential is written $x \Rightarrow y$, and defined as an adjoint functor:

$$z \leq (x \Rightarrow y) \text{ if and only if } z \wedge x \leq y$$

In other words, $x \Rightarrow y$ is a least upper bound for all those elements z with $z \wedge x \leq y$. As a concrete example, for a topological space X the set of open sets $\mathcal{O}(X)$ is a Heyting algebra. The “law of the excluded middle”, meaning $\neg x \vee x = \mathbf{true}$, does not always hold in a Heyting algebra.

Theorem 13. *For any TCM category defined as $\hat{\mathcal{C}} = \mathbf{Sets}^{\mathcal{C}^{op}}$ by the Yoneda embedding $\mathcal{Y}(c)$ of a small causal category $\mathcal{C}$, the partially ordered set $\text{Sub}_{\hat{\mathcal{C}}}(P)$ of subobjects generated by causal interventions on any causal functor defined by the presheaf P is a Heyting algebra.*

Proof: This result follows directly from the corresponding result for any category of presheaves, and is based on constructing the complete lattice $\text{Sub}(P)$ of all subfunctions of P using a pointwise operation for each object $c \in \mathcal{C}$, which can be shown to satisfy an infinite distributive law. $\square$

Causal Mitchell-Bénabou Language and its Kripke-Joyal Semantics

The Causal Mitchell-Bénabou language (CMBL) is a typed local set theory whose syntax and semantics is defined using the arrows of the $\mathcal{C}_{TCM}$ topos. The types of CMBL as causal model objects M of $\mathcal{C}_{TCM}$. For each type M , we assume the existence of variables $x_M, y_M, \dots$, where each such variable has as its interpretation the identity arrow $\mathbf{1} : M \rightarrow M$. We can construct product objects, such as $A \times B \times C$, where terms like σ that define arrows are given the interpretation $\sigma : A \times B \times C \rightarrow D$.

- Each variable x_M of type M is a term of type M , and its interpretation is the identity $x_M = \mathbf{1} : M \rightarrow M$. Here, M may represent an entire SCM, an individual variable, or a causal functor mapping a Markov category to the cocomplete **Prob** category.
- Terms σ and τ of types C and D that are interpreted as $\sigma : A \rightarrow C$ and $\tau : B \rightarrow D$ can be combined to yield a term $\langle \sigma, \tau \rangle$ of type $C \times D$, whose joint interpretation is given as $\langle \sigma p, \tau q \rangle : X \rightarrow C \times D$, where X has the required projections $p : X \rightarrow A$ and $q : X \rightarrow B$. A causal intervention modeled as an arrow $f : X \rightarrow Y$ in $\mathcal{C}_{TCM}$ can be composed with a term $\sigma : U \rightarrow X$ to yield a term of type Y as $f \circ \sigma : U \xrightarrow{\sigma} X \xrightarrow{f} Y$.
- Terms of type Ω are defined as *formulae* of CMBL and can be combined with the usual logical connectives $\wedge, \vee, \Rightarrow, \neg$ and quantifiers $\forall, \exists$ to obtain further terms again of type Ω . An expression such as $\forall x \psi(x, y)$ is interpreted by an arrow $Y \rightarrow \Omega$ (since x is not a free variable). A

formula $\psi(x, y)$ in the topos $\mathcal{C}_{TCM}$ is defined to be *universally valid* in the topos if the corresponding arrow $\psi(x, y) : X \times Y \rightarrow \Omega$ factors through $\mathbf{true} : 1 \rightarrow \Omega$. A formula ψ without free variables is interpreted as an arrow $\psi : 1 \rightarrow \Omega$ and is valid if it coincides with the arrow $\mathbf{true} : 1 \rightarrow \Omega$.

- Indirect proofs (i.e., *reductio ad absurdum*) cannot be used in CMBL because the rule of the excluded middle $\psi \vee \neg\psi$ is not in general valid, nor is the axiom of choice generally true. Instead, the rules of intuitionistic predicate calculus need to be used.
- The Kripke-Joyal semantics for CMBL is specified using *generalized elements*. We define an element of a causal model by the morphism $x : 1 \rightarrow M$. Thus, a generalized element $\alpha : N \rightarrow M$ represents the “stage of definition” of M by N . We specify the semantics of how an TCM model N supports any formula $\phi(\alpha)$, denoted by $N \Vdash \phi(\alpha)$ by $N \Vdash \phi(\alpha)$ if and only if $\text{Im } \alpha \leq \{x | \phi(x)\}$. Stated in the form of a commutative diagram, this “forcing” relationship holds if and only if α factors through $\{x | \phi(x)\}$.

$$\begin{array}{ccccc}
 & & \{x | \phi(x)\} & \xrightarrow{\quad} & 1 \\
 & \nearrow \text{dashed} & \downarrow & & \downarrow \mathbf{true} \\
 N & \xrightarrow{\alpha} & M & \xrightarrow{\phi(x)} & \Omega
 \end{array}$$

Summary

In this chapter, we introduced Topos Causal Models as a categorical envelope for causal presentations. Three topos structures are central: (i) a subobject classifier for predicates and admissible submodels, (ii) an intuitionistic internal logic for stage-dependent truth, and (iii) sheaf conditions for gluing compatible local data. These are additional expressive resources, not a proof that arbitrary subobjects are interventions or that topos semantics alone identifies causal effects. The universal factorization is the standard Yoneda free-cocompletion theorem applied to a causal semantic functor.

The next chapter studies j -do-calculus (“judo calculus”) and its use in decentralized causal discovery. A map–reduce implementation exposes parallelism across regimes, but whether it improves accuracy or wall-clock time is an empirical question rather than a consequence of the topos semantics. The construction uses a Lawvere–Tierney topology $j : \Omega \rightarrow \Omega$ on the subobject classifier to express the relevant internal modality.

We briefly described the Mitchell–Bénabou language that provides the internal language of a topos, along with its Kripke-Joyal semantics. In the next chapter, we will describe it in more detail. A rigorous account of topos

theory, along with the theory of sheaves, and its internal logic is given in this classic book.²⁴

Suggested Reading

The classical foundations of sheaves, topoi, and internal logic are developed by Mac Lane and Moerdijk [1992]. The categorical causal setting is introduced in Mahadevan [2023], while Fritz [2020] provides the complementary Markov-category treatment of probability and conditional independence.

²⁴ Saunders Mac Lane and Ieke Moerdijk. *Sheaves in Geometry and Logic a First Introduction to Topos Theory*. Springer New York, New York, NY, 1992. ISBN 9781461209270 1461209277. URL <http://link.springer.com/book/10.1007/978-1-4612-0927-0>

Judo Calculus

In this chapter, we will introduce j -do-calculus, a powerful sheaf-theoretic framework for decentralized causal discovery (aka “judo” calculus).²⁵ Judo calculus is based on the Lawvere-Tierney topology defined on the subobject classifier in a topos.

JUDO CALCULUS is more formally defined as j -stable causal inference using j -do-calculus in a topos of sheaves $\mathbf{Sh}_j(\mathcal{C})$. A Lawvere–Tierney operator j specifies a modality on internal truth, and j -stability requires the relevant conditional-independence or interventional statement to be closed under that modality. The three rules proved in the companion theory paper are sound relative to their stated internal Markov and forcing semantics. With the identity Lawvere–Tierney operator they reduce to the unmodalized internal rules; a classical Boolean specialization additionally requires a Boolean ambient semantics, such as the ordinary set-based case. The trivial topology alone does not make an arbitrary presheaf topos Boolean.

From Classical Do-Calculus to j -Do-Calculus

In this chapter, we describe how to generalize classical do-calculus to j -stable causal inference inside a topos of sheaves $\mathbf{Sh}_j(\mathcal{C})$, where regimes form a site $(\mathcal{C}, j)$ and observations/interventions are sheaves on that site.

We build on Topos Causal Models from the previous chapter, but impose an explicit admissibility requirement: only structure-preserving subobjects that encode the declared mechanism replacement are interpreted as interventions. A Lawvere–Tierney topology supplies the modal operator j on Ω . The resulting rule system has premises and conclusions in the internal logic of $\mathbf{Sh}_j(\mathcal{C})$, and its soundness is conditional on the internal Markov and forcing assumptions stated in the rule theorems. The identity modality recovers the unmodalized internal rules; the ordinary classical rules are recovered only after a Boolean set-based specialization. Nontrivial covers provide the regime-aware interpretation.

²⁵ Sridhar Mahadevan. Intuitionistic j -do-calculus in topos causal models, 2025c. URL <https://arxiv.org/abs/2510.17944>; and Sridhar Mahadevan. Decentralized causal discovery using judo calculus, 2025b. URL <https://arxiv.org/abs/2510.23942>

Classical Do-Calculus

We briefly review the notion of a structural causal model (SCM), and the classical notion of do-calculus.²⁶ Succinctly, any SCM M defines a unique function from exogenous variables to endogenous variables, and do-calculus models interventions as “sub-functions”:

Definition 56. A **structural causal model (SCM)** is defined as the triple $\langle U, V, F \rangle$ where $V = \{V_1, \dots, V_n\}$ is a set of endogenous variables, U is a set of exogenous variables, F is a set $\{f_1, \dots, f_n\}$ of “local functions” $f_i : U \cup (V \setminus V_i) \rightarrow V_i$ whose composition induces a unique function F from U to V .

Definition 57. Let $M = \langle U, V, F \rangle$ be a causal model defined as an SCM, and X be a subset of variables in V , and x be a particular realization of X . A **submodel** $M_x = \langle U, V, F_x \rangle$ of M is the causal model $M_x = \langle U, V, F_x \rangle$, where $F_x = \{f_i : V_i \notin X\} \cup \{X = x\}$.

Definition 58. Let M be an SCM, X be a set of variables in V , and x be a particular realization of X . The **effect** of an action $\text{do}(X = x)$ on M is given by the submodel M_x .

Definition 59. Let Y be a variable in V , and let X be a subset of V . The **potential outcome** of Y in response to an action $\text{do}(X = x)$, denoted $Y_x(u)$, is the solution of Y for the set of equations F_x .

We can also model counterfactuals in the following way:

Definition 60. Let Y be a variable in V and let X be a subset of V . The counterfactual sentence “The value that Y would have obtained had X been set to x ” is defined as the potential outcome $Y_x(u)$.

Do-calculus provides three algebraic rules for manipulating interventional expressions of the form $P(Y \mid \text{do}(Z), X, W)$ based on conditional independence statements in a causal graph G . The notation used is as follows: $G_{\bar{X}}$ means delete all arrows into X (surgical intervention on X); $G_{\bar{Z}}$: delete all arrows out of Z ; $Z(W)$ denotes the subset of Z that are not ancestors of any node in W in $G_{\bar{X}}$; and finally, $G_{\bar{X}, \overline{Z(W)}}$ denotes the intervention that deletes arrows into those Z -nodes that are not ancestors of W .

1. **Rule 1 (Insertion/Deletion of Observations).** If $(Y \perp Z \mid X, W)_{G_{\bar{X}}}$, then

$$P(Y \mid \text{do}(X), Z, W) = P(Y \mid \text{do}(X), W).$$

2. **Rule 2 (Action/Observation Exchange).** If $(Y \perp Z \mid X, W)_{G_{\bar{X}, \bar{Z}}}$, then

$$P(Y \mid \text{do}(X), \text{do}(Z), W) = P(Y \mid \text{do}(X), Z, W).$$

3. **Rule 3 (Insertion/Deletion of Actions).** If $(Y \perp Z \mid X, W)_{G_{\bar{X}, \overline{Z(W)}}}$, then

$$P(Y \mid \text{do}(X), \text{do}(Z), W) = P(Y \mid \text{do}(X), W).$$

²⁶ Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, USA, 2nd edition, 2009. ISBN 052189560X

Pearl’s rules are sound, and they are complete for the standard problem of identifying causal effects from observational distributions in a causal DAG. This completeness statement concerns identifiability under the graphical-model assumptions; it is not a claim that every equality between arbitrary causal models follows from the three rules.

j-do-Calculus: A Birds-Eye View

In moving from classical do-calculus to j -do-calculus, we place graph-based causal presentations inside a topos. The arrow category of sets is the functor category $\mathbf{Set}^{[1]}$ and is therefore a topos, but this observation alone does not prove that an arbitrary constrained category of SCMs is a topos. A directed graph can likewise be represented as a presheaf on the walking parallel pair: the two structure maps return the source and target of each edge. More generally, for a small category $\mathcal{C}$, the presheaf category $\mathbf{Set}^{\mathcal{C}^{op}}$ is a topos and the Yoneda embedding sends c to $\mathcal{C}(-, c)$. This supplies a useful semantic envelope for causal presentations; causal interpretation and intervention admissibility remain additional structure.

j -DO-CALCULUS rules are summarized in the following figure, and will be explained in more detail in this chapter. At the core of j -do-calculus (aka “judo calculus), the original Grothendieck topology used in TCMs is generalized to use the Lawvere-Tierney j -operator on the subobject classifier Ω .

Causal Models Over a Topos of Sheaves

The categorical framework underlying Topos Causal Models (TCMs) introduced in the previous chapter defines causal inference in a topos by using the property that Yoneda embeddings of presheaves forms a topos. To ensure consistent extension into a unique global function, we build on the theory of sheaves, which ensures local functions can be “collated” together to yield a unique global function. In our setting, we will construct sheaves from categories over causal models through the Yoneda embedding $\mathcal{Y}(x) : \mathcal{C} \rightarrow \mathbf{Sets}^{\mathcal{C}^{op}}$ and impose a Grothendieck topology. TCMs were originally defined over Grothendieck topologies on categories (known as sites), and we will generalize that formulation here to Lawvere-Tierney topologies.

Lawvere-Tierney Topologies on a Topos

In the previous chapter defining TCM, the category of sheaves defining $\mathcal{C}_{TCM}$ (e.g., sheaves over a Markov category) was given a Grothendieck topology. A more elegant framework is to use the Lawvere-Tierney topology on the subobject classifier Ω .

Symbol	Type	Meaning / Typical usage
$\mathbf{C}$	category	Site of “regimes/contexts” (objects are stages; arrows are refinements).
j	L–T topology	Lawvere–Tierney topology on $\mathbf{Sets}^{\mathbf{C}^{op}}$; enforces which sieves are “covering.”
$\mathbf{Sh}_j(\mathbf{C})$	topos	Sheaves on $\mathbf{C}$ for j (the j -reflective subtopos).
a_j	functor	j -sheafification (left exact reflector $\mathbf{Sets}^{\mathbf{C}^{op}} \rightarrow \mathbf{Sh}_j(\mathbf{C})$).
$\mathcal{U}$	object	A stage (or (object) $U \in \mathcal{C}$) in $\mathbf{C}$.
$\mathcal{S} = \{S_i \hookrightarrow \mathcal{U}\}$	family	A j -cover of $\mathcal{U}$ (local charts that jointly “see” $\mathcal{U}$).
$\Vdash_j$	relation	Internal forcing in $\mathbf{Sh}_j(\mathbf{C})$; $\mathcal{U} \Vdash_j \varphi$ reads “ φ holds j -stably at $\mathcal{U}$.”
$X \perp\!\!\!\perp Y \mid Z$	formula	Conditional independence assertion (CI).
$\text{do}(x)$	term	Pearl’s do-operator (surgical intervention) internalized in $\mathbf{Sh}_j(\mathbf{C})$.
$\mathcal{G}_{\overline{X}}$	graph	Mutilated graph with incoming edges to X cut (intervening on X).
$\mathcal{G}_{\underline{Z}}$	graph	Graph with outgoing edges from Z cut (treating Z as “measurement”).
$P(\cdot)$	object	Internal probability in $\mathbf{Sh}_j(\mathbf{C})$; e.g., $P(y \mid \text{do}(x), z, w)$.

Table 24: **Glossary of symbols**

and notation. Informal reading:

j specifies which families of local charts count as covers; $\mathcal{U} \Vdash_j \varphi$ means *every* chart in a j -cover of $\mathcal{U}$ validates φ , hence φ is forced globally at $\mathcal{U}$.

Definition 61 (Lawvere–Tierney causal topology). *Let $\mathbb{E}$ be an elementary topos with subobject classifier Ω and distribution monad $\text{Dist}_{\mathbb{E}}$. A causal topology on $\mathbb{E}$ is a Lawvere–Tierney topology $j : \Omega \rightarrow \Omega$ satisfying:*

$$j(\top) = \top, \quad j(p \wedge q) = j(p) \wedge j(q), \quad j(j(p)) = j(p),$$

where $\top = \mathbf{true}$.

The original Grothendieck topology formulation is a special case of this more general formulation, which we expand on in greater depth in the remainder of the chapter. In particular, we have the following result.

Theorem 14. *If $\mathcal{C}$ is a small category, the Grothendieck topologies J on $\mathcal{C}$ correspond exactly to Lawvere–Tierney topologies on the presheaf topos $\mathbf{Sets}^{\mathcal{C}^{op}}$.*

The above figure gives a diagrammatic illustration of the relationship between the two approaches.

Kripke-Joyal Semantics for Sheaves

Every topos has an internal intuitionistic logic that derives from the fact that the subobject classifier Ω yields a poset of subobjects on which the semantics of a formal Mitchell–Bénabou language describing objects and arrows in the category can be defined. This formal language is associated with a Kripke-Joyal semantics, which we will specialize to a topos equipped with

j-do rules at a glance

All equalities are identities *internal* to $\mathbf{Sh}_j(\mathbf{C})$ and read at stage $\mathcal{U}$ (i.e. under $\mathcal{U} \Vdash_j \dots$). Each premise means: there exists a *j*-cover $\mathcal{S} = \{S_i \rightarrow \mathcal{U}\}_i$ such that the stated CI holds on *every* chart S_i after the indicated graph surgery.

[*j*-Rule 1: insert/delete observations]

$$\left(Y \perp Z \mid X, W \text{ in } \mathcal{G}_{\overline{X}} \text{ on a } j\text{-cover of } \mathcal{U} \right) \implies P(y \mid \text{do}(x), z, w) = P(y \mid \text{do}(x), w).$$

Reading: After cutting arrows into X , if every chart blocks Z from Y given X, W , then observing Z is irrelevant under $\text{do}(x)$.

[*j*-Rule 2: action/observation exchange]

$$\left(Y \perp Z \mid X, W \text{ in } \mathcal{G}_{\overline{X}, \underline{Z}} \text{ on a } j\text{-cover of } \mathcal{U} \right) \implies P(y \mid \text{do}(x), \text{do}(z), w) = P(y \mid \text{do}(x), z, w).$$

Reading: After cutting arrows into X and *out of* Z , intervening on Z equals observing Z under $\text{do}(x)$, chartwise.

[*j*-Rule 3: insert/delete actions]

$$\left(Y \perp Z \mid X, W \text{ in } \mathcal{G}_{\overline{X}, \overline{Z(W)}} \text{ on a } j\text{-cover of } \mathcal{U} \right) \implies P(y \mid \text{do}(x), \text{do}(z), w) = P(y \mid \text{do}(x), w).$$

Reading: After cutting arrows into X and into the parents of Z not in W (i.e. $\overline{Z(W)}$), if every chart blocks Z from Y given X, W , then $\text{do}(z)$ is irrelevant under $\text{do}(x)$.

Figure 39: The Rules of *j*-do-calculus.

$$\begin{array}{ccc} \mathbf{Sh}(\mathcal{C}, J) & \xrightarrow{\quad \cong \quad} & \mathcal{E}_j \\ & \searrow i_j \quad \swarrow i_j & \\ & [\mathcal{C}^{\text{op}}, \text{Set}] & \end{array}$$

Figure 40: External Grothendieck topology J and internal Lawvere–Tierney topology j both induce subtopoi embedded in the presheaf topos $[\mathcal{C}^{\text{op}}, \text{Set}]$.

a Grothendieck topology, that is a site. This specialized structure captures how causal inference is woven in the fabric of the internal logic of a causal topos. Define $\mathbf{Sh}(\mathcal{C}, \mathcal{J})$ be a topos of sheaves with a specified Grothendieck topology $\mathcal{J}$, defined by the following diagram, where $\mathcal{Y}$ is the Yoneda embedding, and $\mathcal{P}$ is a presheaf:

$$\mathcal{C} \xrightarrow{\mathcal{Y}} \mathcal{P}(\mathcal{C}) \xrightarrow{a} \mathbf{Sh}(\mathcal{C}, \mathcal{J}) \cong \mathcal{C}$$

where we know that the Yoneda embedding $\mathcal{Y}$ creates a full and faithful copy of the original category $\mathcal{C}$. Let us define the semantics for a sheaf element $\alpha \in X(\mathcal{C})$, where $X(\mathcal{C}) = \mathbf{Sh}(\mathcal{C}, J)(\mathcal{C}(-, \mathcal{C}), X)$. We will describe the Kripke–Joyal semantics in more detail later in the chapter, but for now, a concise summary for the topos category of sheaves is as follows:

1. $\mathcal{C} \Vdash \phi(\alpha) \wedge \psi(\alpha)$ if it holds that $\mathcal{C} \Vdash \phi(\alpha)$ and $\mathcal{C} \Vdash \psi(\alpha)$.

2. $C \Vdash \phi(\alpha) \vee \psi(\alpha)$ if there is a covering $\{f_i : C_i \rightarrow C\}$ such that for each i , either $C_i \Vdash \phi(\alpha)$ or $C_i \Vdash \psi(\alpha)$.
3. $C \Vdash \phi(\alpha) \rightarrow \psi(\alpha)$ if for all $f : D \rightarrow C$, and $D \Vdash \phi(\alpha \circ f)$, it holds that $D \Vdash \psi(\alpha \circ f)$.
4. $C \Vdash \neg\phi(\alpha)$ holds if for all arrows $f : D \rightarrow C$ in $\mathcal{C}$, if $D \Vdash \phi(\alpha \circ f)$ holds, then the empty family is a cover of D .
5. $C \Vdash \exists y \phi(x, y)$ holds if there is a covering $\{f_i : C_i \rightarrow C\}$ and elements $\beta_i \in Y(C_i)$ such that $C_i \Vdash \phi(\alpha \circ f_i, \beta_i)$ holds for each i .
6. Finally, for universal quantification, $C \Vdash \forall y \phi(x, y)$ holds if for all arrows $f : D \rightarrow C$ in the category $\mathcal{C}$, and all $\beta \in Y(D)$, it holds that $D \Vdash \phi(\alpha \circ f, \beta)$.

j-do-Calculus on Sites

To transition from classical do-calculus to *j*-do-calculus, we need to provide a “bridge” that maps from classical notions, like d-separation, to intuitionistic notions in *j*-do-calculus. We begin this transition by introducing some terms that will be used in the remainder of the chapter.

STAGES AND GENERALIZED ELEMENTS. Let $(\mathcal{C}, J)$ be a site and $\mathbf{Sh}_J(\mathcal{C})$ its sheaf topos. For any object $A \in \mathbf{Ob}(\mathcal{C})$, a *generalized element of A at stage V* is a morphism $f : V \rightarrow A$ (equivalently, an element of the presheaf $yA(V) = \text{Hom}_{\mathcal{C}}(V, A)$). The special case $\mathbf{1} \rightarrow A$ (where $\mathbf{1}$ is terminal) is a *global element*. In what follows we fix an *ambient context* (or *ambient object*) $U \in \mathbf{Ob}(\mathcal{C})$ and call any arrow $f : V \rightarrow U$ a *local stage over U* .

CHARTS (“REGIMES”) AND J -COVERS. A *chart* (our earlier “regime”) is precisely a local stage $f : V \rightarrow U$. A family of charts $\{f_i : V_i \rightarrow U\}_{i \in I}$ generates the sieve

$$\langle f_i \rangle = \{h : W \rightarrow U \mid \exists i, \exists g : W \rightarrow V_i \text{ with } h = f_i \circ g\}.$$

We call $\{f_i\}$ a *J -cover of U* iff $\langle f_i \rangle \in J(U)$ (i.e. the generated sieve is *J -covering*).

READING FORMULAS “AT STAGE U ”. Let φ be a formula in the internal language. Write $U \Vdash_J \varphi$ to mean that φ is (internally) true at the ambient object U in $\mathbf{Sh}_J(\mathcal{C})$. In Kripke–Joyal semantics this is equivalent to the existence of a J -covering sieve $S \subseteq \text{Hom}_{\mathcal{C}}(-, U)$ such that each local stage $f : V \rightarrow U$ in S forces φ after pullback:

$$U \Vdash_J \varphi \iff \exists S \in J(U) \text{ with } \forall f : V \rightarrow U \text{ in } S, \quad V \Vdash_J \varphi|_f.$$

Informally: φ holds chartwise on a J -cover of U .

GROTHENDIECK TOPOLOGY AND J -COVERS. A *sieve* S on U is J -*covering* iff $S \in J(U)$. We will say that a family of charts $\{f_i: V_i \rightarrow U\}$ is a J -*cover* of U iff the sieve it generates is J -covering:

$$\{f_i\} \text{ is a } J\text{-cover of } U \iff \langle f_i \rangle \in J(U).$$

Thus our earlier “ J -cover” phrase always refers to a *covering family whose generated sieve is J -covering*.

LAWVERE–TIERNEY TOPOLOGY j AND J . The Grothendieck topology J on $\mathcal{C}$ corresponds to a Lawvere–Tierney topology $j: \Omega \rightarrow \Omega$ on the presheaf topos $\widehat{\mathcal{C}}$; the sheaf topos $\mathbf{Sh}_J(\mathcal{C})$ is the j -sheaf subtopos of $\widehat{\mathcal{C}}$. We freely pass between J (external/topological) and j (internal/logical) viewpoints; “ j -closure” of a subobject corresponds to saturation under J -covering sieves. **Slogan.** A conditional independence $\varphi \equiv (X \perp\!\!\!\perp Y \mid Z)$ is j -*stable* at a stage $\mathcal{U}$ iff the sieve of all refinements $u: V \rightarrow \mathcal{U}$ that validate φ is a J -cover of $\mathcal{U}$.

SITE OF CAUSAL CONTEXTS. Fix a finite variable set $\mathcal{V}$ and a DAG G on $\mathcal{V}$. A *stage* is a pair $\mathcal{U} = (G, \sigma)$, where σ is a status profile that records which nodes are (i) conditioned/observed, (ii) intervened upon (incoming arrows cut), etc. A *morphism* $u: (G', \sigma') \rightarrow (G, \sigma)$ is a refinement that is identity on node names and *monotone in status* (a refinement may condition or intervene on more variables, but never less). Stages and refinements form a category $\mathbf{C}$.

OPEN PATHS AND SATISFACTION. For disjoint $X, Y, Z \subseteq \mathcal{V}$ and a stage $\mathcal{U} = (G, \sigma)$, let $\text{OpenPaths}_{\mathcal{U}}(X, Y \mid Z)$ be the set of G -paths from X to Y that are d -open under the usual collider/non-collider rules, evaluated after applying the surgeries in σ (e.g., $\text{do}(\cdot)$). Write

$$\mathcal{U} \models (X \perp\!\!\!\perp Y \mid Z) \iff \text{OpenPaths}_{\mathcal{U}}(X, Y \mid Z) = \emptyset.$$

THE SIEVE SELECTED BY A CI FORMULA. Given $\varphi \equiv (X \perp\!\!\!\perp Y \mid Z)$ and $\mathcal{U}$, define

$$S_{\varphi}(\mathcal{U}) := \{u: V \rightarrow \mathcal{U} \text{ in } \mathbf{C} \mid V \models \varphi\}.$$

Lemma (sieve). $S_{\varphi}(\mathcal{U})$ is a sieve on $\mathcal{U}$ (i.e., closed under precomposition).

Proof sketch. If $u: V \rightarrow \mathcal{U}$ validates φ and $w: W \rightarrow V$ is any arrow, then W refines V monotonically in status, which can only block additional paths; hence $W \models \varphi$ and $u \circ w \in S_{\varphi}(\mathcal{U})$. $\square$

GROTHENDIECK TOPOLOGIES FROM ADMISSIBLE CHARTS. Fix for each $\mathcal{U}$ a family $\{\rho_k: V_k \rightarrow \mathcal{U}\}_{k \in K}$ of *admissible local views* (charts) used

to test CI at $\mathcal{U}$ (e.g., purely observational; or a mix including certain $\text{do}(\cdot)$ -surgeries). Let J be the Grothendieck topology *generated* by these bases: a sieve S covers $\mathcal{U}$ iff it contains a jointly epimorphic family refining $\{\rho_k\}$. Two canonical choices:

- J_{id} (classical): basis = $\{\text{id}_{\mathcal{U}}\}$.
- J_{mix} : basis includes observational charts and specific interventional charts.

FORCING SEMANTICS (j -STABILITY). Write

$$\mathcal{U} \Vdash_J (X \perp\!\!\!\perp Y \mid Z) \quad :\Longleftrightarrow \quad S_\varphi(\mathcal{U}) \text{ is a } J\text{-cover of } \mathcal{U}.$$

Proposition (conservativity). With J_{id} ,

$$\mathcal{U} \Vdash_{J_{\text{id}}} (X \perp\!\!\!\perp Y \mid Z) \quad \Longleftrightarrow \quad \mathcal{U} \models (X \perp\!\!\!\perp Y \mid Z).$$

Reason. A sieve covers $\mathcal{U}$ in J_{id} iff it contains $\text{id}_{\mathcal{U}}$. Thus $S_\varphi(\mathcal{U})$ covers iff $\text{id}_{\mathcal{U}} \in S_\varphi(\mathcal{U})$, i.e., $\mathcal{U} \models \varphi$. $\square$

Proposition (soundness of j -stability). If $\{\rho_k : V_k \rightarrow \mathcal{U}\}$ generates J at $\mathcal{U}$ and $V_k \models (X \perp\!\!\!\perp Y \mid Z)$ for all k , then $\mathcal{U} \Vdash_J (X \perp\!\!\!\perp Y \mid Z)$.

Reason. Each generator ρ_k lies in $S_\varphi(\mathcal{U})$; hence the sieve they generate covers, and by upward closure of covering sieves, so does $S_\varphi(\mathcal{U})$. $\square$

WORKED MAPPING: EARTHQUAKE EXAMPLE. Let $\mathcal{U} = (G, \sigma)$ with $B \rightarrow A \leftarrow E$ and $A \rightarrow C$. Take J_{mix} generated by two charts: an *observational* chart ρ_{obs} (no conditioning on colliders unless stated) and an *interventional* chart $\rho_{\text{do}A}$ that cuts the incoming edges into A . Then:

- (i) $\mathcal{U} \Vdash_{J_{\text{mix}}} (B \perp\!\!\!\perp E)$ (collider closed in obs; parents cut under $\text{do}(A)$).
- (ii) $\mathcal{U} \Vdash_{J_{\text{mix}}} (B \perp\!\!\!\perp C \mid A)$ (chain blocked by A in both charts).
- (iii) $\mathcal{U} \not\Vdash_{J_{\text{mix}}} (B \perp\!\!\!\perp E \mid A)$ (conditioning on the collider opens the path in the obs chart).

TAKEAWAY. A CI formula φ determines a sieve S_φ ; a Grothendieck topology J encodes which local views count as *covers*. Classical CI is truth at $\mathcal{U}$; j -stability is truth on a J -cover of $\mathcal{U}$ —i.e., gluable from admissible local regimes.

CI AS AN INTERNAL PREDICATE. Fix a graph object G (DAG with surgery) represented in $\mathcal{C}$. For variables X, Y, Z (as objects/indices in G), let $\perp\!\!\!\perp_G (X; Y \mid Z)$ denote the internal formula “ $X \perp\!\!\!\perp Y \mid Z$ in G ”. Our usage

$$“Y \perp\!\!\!\perp Z \mid X, W \text{ in } \tilde{G}^{(\cdot)} \text{ on a } J\text{-cover of } \mathcal{U}”$$

means precisely: there exists a J -covering sieve $S \subseteq \text{Hom}(-, \mathcal{U})$ such that for every $f : V \rightarrow U$ in S , the (pulled-back, surgically modified) graph satisfies $V \Vdash_J \perp\!\!\!\perp_G (X; Y \mid Z)$. By the clause above, this suffices to conclude $\mathcal{U} \Vdash_J \perp\!\!\!\perp_G (X; Y \mid Z)$.

Characteristic	Classical do-calculus	Judo Calculus
Logic	Boolean: causal claims are globally true or false	Intuitionistic logic: truth is <i>local</i>
Context	Uses “average” treatment effect	Local truth is “glued” together
Interventions	“Surgery” of a causal graph	Subobject classifier
Identification	Axioms define three rules	More general axiomatic framework

Table 25: Some of the salient differences between classical do-calculus and judo calculus.

Algorithms for Judo Calculus

The algorithmic framework for judo calculus combines it with standard score-based, constraint-based, and gradient-based causal discovery methods [Zanga and Stella, 2023]. Its experiments form data-driven j -covers, compute chartwise conditional independences after graph surgeries, and glue them to certify the premises of the j -do rules in practice. They compare ordinary and j -stable variants of causal-discovery methods on synthetic data and real-world datasets from biology and economics; the original study gives the complete experimental specification [Mahadevan, 2025b].

We first give a high-level overview of judo calculus in this section, prior to introducing the highly technical definitions on which it is rigorously based. Judo calculus is *intuitionistic*, meaning that the law of the excluded middle $p \vee \neg p$ does not necessarily hold, and $\neg\neg p$ cannot be reduced to p , for any causal proposition p . Rather, judo calculus requires a constructive proof to determine a statement’s truth. In the context of causality, this means you can’t assume that a causal claim is either universally true or universally false without a proof for one or the other.

Judo calculus uses topos theory to model causality in a more flexible way than classical methods. For example, in many real-world applications of causal inference, a particular intervention, such as administering a drug or employing a lunch program in schools, may not be effective over the entire population, but rather in different “regimes” (e.g., senior citizens may respond more favorably than younger recipients, and similarly, students from low-income backgrounds may benefit nutritionally from a free lunch program than students from high-income regions). The following table summarizes the differences between classical do-calculus and judo calculus.

As a concrete example, let us imagine that a city planning a public health initiative to combat childhood obesity decides to distribute free healthy lunches in public schools. The city policy makers want to determine if this intervention reduces the students’ body mass index (BMI) by the year’s end. Classical do-calculus would seek a global average treatment effect over the entire school population. Judo calculus makes it possible to define individual regimes, such as “low-income” and “high-income” where the causal intervention may or may not be as effective.

Judo calculus works in the setting of sheaves or sites, which are categories that are equipped with a Lawvere-Tierney topology defined by a modal operator j on the subobjects. In plain English, this means that the category has a

topology defined by the arrows, and their compositional structure. This j operator defines a notion of causal “stability”, which will be studied extensively in the coming sections. For example, in the “low income” students, the city’s causal intervention may be j -stable, whereas in the “high income” category, this intervention may not be as effective. The operator j acts on the subobject classifier Ω : an object in the category that serves to define “truth”, which in general is not Boolean. The j operator acts to determine “local” truth from “global” truth, and also provides a closure property to determine j -stability. The j operator on Ω is defined to represent the notion of a “globally valid” causal statement. A j -cover represents a tessellation of the space of arrows that all have a common co-domain.

Judo Calculus Model of Causal Inference under Interference

In this section, we want to quickly show how our framework applies to the case when the standard Stable Treatment Unit Value Assumption (SUTVA) is violated because of “interference”. For example, a report on a study involving power plants that may cause pollution among the residents of a particular county, where each individual resident may be exposed to the radiation from multiple power plants. A particular treatment on a power plant, such as imposing pollution controls, may affect many residents.

Classical do-calculus causal inference often assumes SUTVA (no interference): a unit’s outcome depends only on its own treatment. In many spatial and networked problems this fails. Consider air pollution: a resident may be exposed to emissions from multiple plants, with exposure modulated by meteorology. Claims that ignore this heterogeneity are either false or too brittle to be useful. Our framework captures this by proving *local* causal truths on families of comparable regimes (*covers*) and then *gluing* them. We illustrate this in a two-source interference model with overlapping covers. Judo calculus shows how to certify claims locally on covers and glue them via j -stability. This simple example shows the power of the judo calculus formalism in modeling a range of real-world examples, and we will get into several real-world datasets in the experiments described later in the chapter.

SETTING. Let $P = \{1, 2\}$ be two sources (plants) and R a population of residents. Time is indexed by t . Each plant has a binary treatment $Z_p(t) \in \{0, 1\}$ (e.g., a scrubber on/off). Meteorology $(\theta(t), M(t))$ denotes wind direction (degrees) and a mixing proxy (e.g., stability or mixing height). Interference arises because multiple sources contribute to each resident’s exposure.

EXPOSURE MAPPING. For each resident r , define wind-dependent weights $w_{rp}(\theta, M) \geq 0$ (larger when r is downwind from p). Write the time-varying

exposure as

$$E_r(t) = \sum_{p \in P} w_{rp}(\theta(t), M(t)) Z_p(t) + \eta_r(t),$$

and model the outcome as

$$Y_r(t) = \beta_1 E_{r1}(t) + \beta_2 E_{r2}(t) + \varepsilon_r(t),$$

where E_{rp} is the contribution traceable to source p . SUTVA is violated by design: Y_r depends on the vector $Z = (Z_1, Z_2)$ through the mapping $E_r(\cdot)$.

REGIMES AND COVERS. A *regime* is a subset of times where the exposure mapping is comparable (e.g., similar wind sector or mixing). Let

WL: west-large (230–310°); WS: west-small (250–290°);

E: east (70–110°); LM: low mixing (e.g., $M < -0.5$).

These regimes *overlap*: we also consider intersections such as $WL \cap LM$ and $E \cap LM$. In practice, we may refine a regime into smaller *charts* (disjoint sub-regimes) to assess within-regime variability.

PRESHEAF VIEWPOINT. Let $\mathcal{R}$ be the (small) category whose objects are regimes ($E, WL, WS, LM, \dots$) and whose morphisms are inclusions (refinements). Define a presheaf $\mathcal{F} : \mathcal{R}^{op} \rightarrow \mathbf{Set}$ that assigns to each regime e the set of *local models or summaries* computable on e (e.g., regression coefficients, edge presence/absence), with restriction maps along inclusions. A *local truth* on e is a property $\varphi \in \mathcal{F}(e)$ we can build constructively from data in e . A *cover* $\mathcal{J} = \{e_i \rightarrow e\}$ is a jointly surjective family of inclusions; φ is *j-stable* on e if $\varphi|_{e_i}$ holds for every $e_i \in \mathcal{J}$ and these local sections are compatible (glue).

OPERATIONAL TEST FOR j-STABILITY (INTERFERENCE). We instantiate $\mathcal{F}$ with simple local regressions and a frequency threshold:

1. **Charts.** For each regime $e \in \{WL, WS, E, LM, WL \cap LM, E \cap LM\}$, split the time indices into K disjoint charts $e = \bigsqcup_{k=1}^K e^{(k)}$ (e.g., equal-size shards).
2. **Local models.** On each chart $e^{(k)}$, fit a standardized OLS model $Y \sim E_1 + E_2$. Record the coefficients $\hat{\beta}_1^{(k)}(e), \hat{\beta}_2^{(k)}(e)$. Declare the edge $E_1 \rightarrow Y$ *present* on the chart if $|\hat{\beta}_1^{(k)}(e)| \geq \tau_\beta$ (e.g., $\tau_\beta = 0.2$); similarly for $E_2 \rightarrow Y$.
3. **Frequencies.** Compute edge frequencies $f_{E_1 \rightarrow Y}(e) = \frac{1}{K} \sum_k \mathbf{1}\{|\hat{\beta}_1^{(k)}(e)| \geq \tau_\beta\}$, and $f_{E_2 \rightarrow Y}(e)$ analogously.

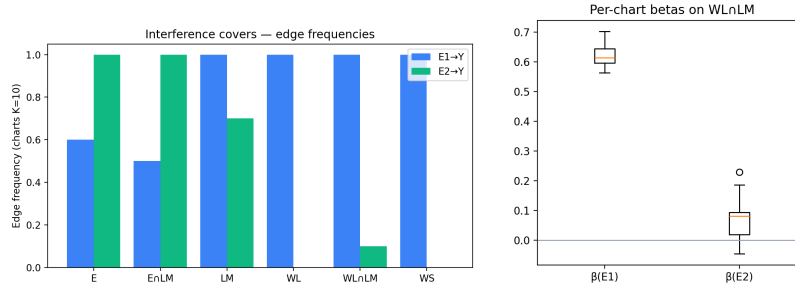


Figure 41: **Interference with overlapping covers.** Edge frequencies $f(E_1 \rightarrow Y), f(E_2 \rightarrow Y)$ by cover (left); per-chart coefficients on intersections (right). Local claims are j -stable on each cover and persist on intersections.

4. **Stability decision.** Fix a stability threshold $\pi \in [0, 1]$. We say $E_p \rightarrow Y$ is j -stable on the cover $\mathcal{J} = \{e_i \rightarrow e\}$ if $f_{E_p \rightarrow Y}(e_i) \geq \pi$ for all $e_i \in \mathcal{J}$. In words: the claim holds on *every chart of every member* of the cover.

The modality j is the Lawvere–Tierney topology that *closes* sieves under the chosen cover: truth is tested along the arrows $e_i \rightarrow e$.

WHAT WE OBSERVE. In a minimal simulation (two plants; wind-dependent weights), the west covers ($WL, WS, WL \cap LM$) yield $f_{E_1 \rightarrow Y} \approx 1.0$ and $f_{E_2 \rightarrow Y} \approx 0$; the east covers ($E, E \cap LM$) flip this, with $f_{E_2 \rightarrow Y} \approx 1.0$ and $f_{E_1 \rightarrow Y} \approx 0.5$ – 0.6 . On LM , both edges are stable (e.g., $f_{E_1 \rightarrow Y} \approx 1.0$, $f_{E_2 \rightarrow Y} \approx 0.7$). Crucially, the same local claims persist on *intersections* (e.g., $WL \cap LM, E \cap LM$), illustrating that our sheaf-based proof obligation naturally handles overlapping charts.

GLUING INTO A STRUCTURE. When one aggregates many regimes, the local edges can be *glued* into a global π -stable skeleton by keeping edges whose frequencies exceed π in every member of the cover and on the intersections. In practice we select π by *validation likelihood* on held-out charts/regimes and then orient edges by a net-preference rule (e.g., keep $i \rightarrow j$ if $f_{i \rightarrow j} - f_{j \rightarrow i}$ exceeds a small margin), optionally imposing domain guards (e.g., forbid edges from the composite to its components).

WHY THIS MATTERS. Instead of a fragile global statement (“reducing emissions at plant 1 lowers hospitalizations everywhere”), we make a j -stable claim that is provably true on the relevant cover (e.g., west-wind regimes and their intersections with low mixing). This separation of concerns—*choose a cover, prove locally, glue globally*—is the essence of the j -do calculus: the usual identification rules apply, but only after claims have been made stable along the cover.

Computational and Statistical Efficiency of j -Stable Discovery

Another significant advantage of judo calculus is its highly decentralized characteristic. A J -cover $\mathcal{S} = \{V_i \hookrightarrow U\}_{i=1}^E$ turns one hard pooled causal discovery problem into E independent subproblems plus a light-weight aggregation. This matches a map–reduce pattern:

$$\underbrace{\text{DISCOVER}(U)}_{\text{pooled}} \rightsquigarrow \left\{ \underbrace{\text{DISCOVER}(V_i)}_{\text{per-env/chart}} \right\}_{i=1}^E \text{ then } \underbrace{\text{GLUE}(\{A_i\})}_{j\text{-aggregation}}.$$

In practice the map phase (per-env FCI/GES/DCDI) is *highly parallel*, and the reduce phase is an $O(Ep^2)$ Boolean fold over adjacency matrices, with short-circuit opportunities (below).

A SIMPLE COST MODEL. Let p be variables, n pooled samples, and n_i samples per chart ($\sum_i n_i = n$). For depth $d=0$ (marginal tests), Fisher-z CI tests are $O(n)$ each, and the skeleton step is $O(p^2n)$.

$$T_{\text{FCI}}^{\text{pooled}}(n, p, 0) \approx c p^2 n,$$

$$T_{\text{FCI}}^{\text{sheaf}}(\{n_i\}, p, 0) \approx \sum_{i=1}^E c p^2 n_i = c p^2 n,$$

so the *sequential* work is comparable. However, with W workers the *wall-clock* time is

$$T_{\text{wall}}^{\text{sheaf}} \approx \max_{i \leq E} c p^2 n_i + O(Ep^2),$$

often much smaller than $T_{\text{FCI}}^{\text{pooled}}$ because: (i) parallelism, (ii) better cache/memory behavior on smaller n_i , and (iii) short-circuiting during aggregation.

For $d>0$, the CI test count grows with the number and size of conditioning sets; the same parallel advantage holds, and smaller n_i usually makes regression/covariance subroutines cheaper and more numerically stable.

SHORT-CIRCUIT AGGREGATION. Let $A_i \in \{0, 1\}^{p \times p}$ be per-chart adjacencies and Agg_τ denote a threshold rule “keep an edge if it appears in at least τ charts.” Intersection and union are special cases with $\tau=E$ and $\tau=1$.

For intersection ($\tau=E$) a single zero in any chart kills the edge (stop early). For union ($\tau=1$) a single one confirms the edge. For k -of- E , both early *accept* and *reject* bounds apply. This makes the reducer essentially linear in the number of *decisive* charts per edge.

MEMORY, I/O, AND PRIVACY. Per-chart runs touch only their own $n_i \times p$ block and emit a compact $p \times p$ binary matrix; the reducer never needs raw data. This also enables data-silo settings: share adjacencies, not samples.

STATISTICAL EFFICIENCY (STABILITY AS REGULARIZATION).

Aggregating across charts is analogous to stability selection and ensemble model averaging:

d	Vanilla	J-stable	J-stable / Vanilla
10	30.3	27.8	0.90
20	51.8	42.3	0.82
40	143.5	101.9	0.71

Table 26: Per-iteration CPU cost (seconds per 10,000 iterations) for vanilla and j -stable DCDI on the recorded synthetic benchmark. Lower is better.

- **Variance reduction.** Spurious edges that appear by chance in one environment are filtered by intersection or by a high support threshold.
- **Bias–variance tradeoff.** Intersection is conservative (low FP, higher FN); k -of- E balances sensitivity and specificity. The support curve (fraction of edges surviving threshold t) is an empirical stability diagnostic.
- **Implicit interventions.** Heterogeneity across charts (different regimes/perturbations) breaks symmetries that pooled data cannot, improving identifiability and power of CI tests.
- **Multiple testing control.** Viewing “edge present in chart i ” as repeated evidence, support thresholds act as a robust filter against per-chart test noise without tuning α aggressively.

Experimental Validation of Judo Calculus Efficiency

The implementation decomposes fitting across regimes and can therefore execute those fits in parallel. The timing plots below record measurements from one synthetic implementation and hardware configuration. They illustrate the available parallelism, but they are not a controlled systems benchmark: the checked artifacts do not establish a general speedup over optimized DCDI.

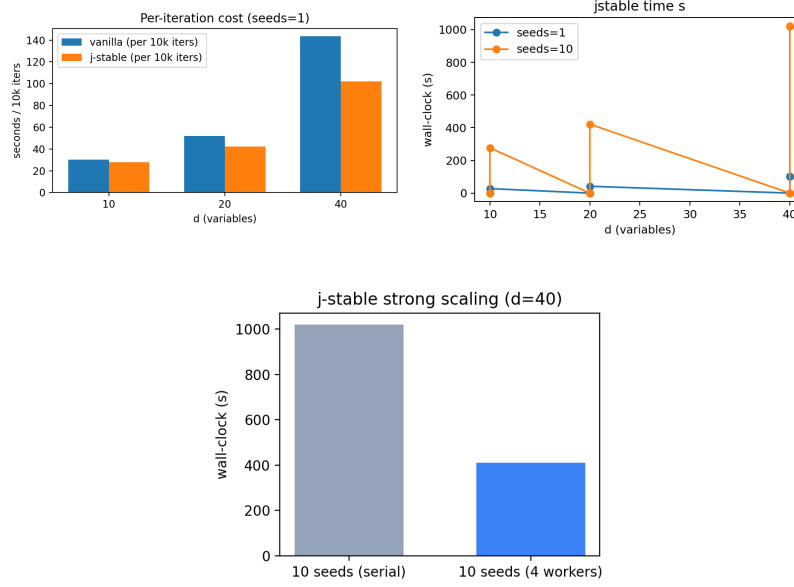


Figure 42: Recorded implementation timings: per-iteration cost (left) and parallel execution of a seed ensemble (right). These are descriptive, not a hardware-controlled comparison with optimized DCDI.

Figure 43: Recorded scaling curve for the checked $d = 40$ synthetic run. The curve shows parallelism in this implementation, not a general complexity separation between the two methods.

The j -stable do-operator (practical form)

Let J be a cover (the family of *comparable* regimes relevant for a query), and let $\{\mathbb{P}^e(\cdot)\}_{e \in J(w)}$ be regime-specific interventional models at covariate value w . We define the j -stable intervention probability by a *monotone* aggregator over the cover:

$$\mathbb{P}_J(Y \in A \mid \text{do}_J(X=x), W=w) := \text{Agg}_{e \in J(w)} \left[\mathbb{P}^e(Y \in A \mid \text{do}(X=x), W=w) \right].$$

Semantics. Choose a cover J (e.g., countries with comparable measurement, QC-passed sites, or a wind sector in interference), evaluate the usual interventional conditional in each regime $e \in J(w)$, and combine them with a pre-registered, order-preserving Agg (Fisher/Stouffer pooling, trimmed mean, etc.). The result is a *local* do-query certified on J .

FROM STRUCTURE TO j -DO. Our discovery layer certifies structure on J via a stability map $F \in [0, 1]^{d \times d}$: an edge $\{i, j\}$ is π -stable if $\max(F_{i \rightarrow j}, F_{j \rightarrow i}) \geq \pi$, and simple margins orient $i \rightarrow j$ when $F_{i \rightarrow j} - F_{j \rightarrow i} \geq \delta$. We select π (and sparsity for baselines) by validation likelihood on held-out regimes, and only then apply the rules of Judo calculus shown earlier.

Remark. The premises “ $\cdot \mid (\cdot, J)$ ” are certified by the π -stable structure (and margins) on the cover. Formal details appear in the companion theory chapter.

EXAMPLE (PISA ESCS). Let J be the set of OECD countries with comparable SES measurement; W include SES deciles; X = instruction time;

Y = math score. Suppose discovery yields a π -stable skeleton/orientation indicating that Z =ESCS satisfies the j -backdoor premise for $X \rightarrow Y$ on J . Then

$$\mathbb{P}_J(Y \leq y \mid \text{do}_J(X=x)) = \text{Agg}_{e \in J} \sum_z \mathbb{P}^e(Y \leq y \mid X=x, Z=z) \mathbb{P}^e(Z=z).$$

In words: evaluate the usual adjustment within each country and aggregate over the cover; the claim is local (on J) and certified by the π -stable structure.

Relation to transportability (Pearl–Bareinboim)

Transportability asks: given a source domain s (e.g., an RCT) and a target domain t , can we express $P_t(y \mid \text{do}(x))$ in terms of source quantities $P_s(\cdot)$ and target observables $P_t(\cdot)$? The method uses selection diagrams and classical do-calculus to derive adjustment/transport formulas; its claims are *global* within the chosen model and focus on identifiability of an effect under population shifts.

J -stability (judo calculus) asks: for a chosen family of comparable regimes (a cover) J , does a causal claim hold *locally and constructively* across the cover? Operationally, we certify a π -stable structure on J (frequency map F , π -skeleton, and net-preference orientation), and then evaluate J -do queries via an order-preserving aggregator:

$$\mathbb{P}_J(Y \in A \mid \text{do}_J(X=x), W=w) := \text{Agg}_{e \in J(w)} \left[\mathbb{P}^e(Y \in A \mid \text{do}(X=x), W=w) \right].$$

HOW THEY CONNECT.

- *Transportability as a special case.* If the cover J contains the source and target ($\{s, t\} \subseteq J$) and we choose Agg to *select* the target component (the identity on t), then $\mathbb{P}_J(y \mid \text{do}_J(x))$ reduces to $P_t(y \mid \text{do}(x))$. When a selection-diagram transport formula holds (e.g., backdoor on Z), the same Z is a J -admissible adjustment set and the J -backdoor rule yields the transported effect.
- *Broader locality.* Transportability typically assumes one global model with population differences mediated by effect modifiers. J -stability allows *covers with intersections* and context-dependent mechanisms: a claim may be certified on wheat-flour sites, winter/wind regimes, or genotype strata without committing to a single universal model.
- *Constructive robustness.* A transported formula says an effect *can* be mapped to t given assumptions. A J -stable claim says the effect *does* hold on the cover J because it is witnessed (e.g., by repeated per-regime fits) and glued by a monotone aggregator; the premises are empirically testable (stability margins, π -support).

TAKEAWAY. Transportability provides *formulas* for source $\rightarrow$ target transfer; J -stability provides a *logic of local truth* and an empirical workflow (stability maps $\Rightarrow J$ -do) that subsumes transport as the two-regime, identity-aggregation case and extends it to richer covers where mechanisms and validity sets vary by context.

Example. Let $J = \{s = \text{RCT}, t = \text{registry}\}$ and suppose Z is J -admissible (stable backdoor). Then $\mathbb{P}_J(y \mid \text{do}_J(x)) = \sum_z \mathbb{P}^t(y \mid x, z) \mathbb{P}^t(z) = P_t(y \mid \text{do}(x))$, which matches the classical transport formula for $(s \rightarrow t)$.

Experimental Validation of j -Stable Causal Discovery

SETUP: REGIMES, PER-REGIME GRAPHS, AND AGGREGATION. Let $V = \{X_1, \dots, X_d\}$ be the variables. We observe data partitioned into E regimes (a.k.a. environments) $\mathcal{E} = \{e_1, \dots, e_E\}$ coming either from known labels (e.g. a column `env`) or from a preprocessing that induces regimes (e.g. clustering). For a causal learner $\mathcal{A} \in \{\text{GES}, \text{PSI-FCI}, \text{DCDI}\}$ and a regime $e \in \mathcal{E}$, we fit $\mathcal{A}$ on the subset data(e) and obtain a (possibly partially oriented) graph summary, represented here as an adjacency matrix

$$A^{(e)} \in \{0, 1\}^{d \times d}, \quad A_{ij}^{(e)} = 1 \iff \mathcal{A} \text{ asserts an edge } X_i \rightarrow X_j \text{ (or an adjacency } X_i - X_j) \text{ in regime } e.$$

From the E matrices we form the *support counts*

$$C_{ij} := \sum_{e \in \mathcal{E}} A_{ij}^{(e)} \in \{0, \dots, E\}, \quad \text{and the stability ratios } s_{ij} := \frac{C_{ij}}{E} \in [0, 1].$$

These are the sufficient statistics of the j -stable aggregation layer. We report three derived graphs:

$$\begin{aligned} A_{ij}^\cap &= \mathbf{1}[C_{ij} = E] && \text{(intersection),} \\ A_{ij}^\cup &= \mathbf{1}[C_{ij} \geq 1] && \text{(union),} \\ A_{ij}^{(k)} &= \mathbf{1}[C_{ij} \geq E - k] && \text{("all-but- k ").} \end{aligned}$$

Optionally, a score threshold $\tau \in (0, 1]$ yields a soft variant $A_{ij}^{(\tau)} = \mathbf{1}[s_{ij} \geq \tau]$.

WHY THIS REALIZES j -STABILITY. In the theory, a conditional statement φ is j -stable at stage U iff there exists a J -cover $\{f_\alpha : V_\alpha \rightarrow U\}$ such that each chart V_α locally validates φ . Here, the regimes play the role of charts: each $e \in \mathcal{E}$ is a refinement of the ambient stage (same node set, refined observation/intervention status), and the learner $\mathcal{A}$ produces *local* edge claims $A_{ij}^{(e)} \in \{0, 1\}$. Intersection $A^\cap$ certifies edges that hold on *all* charts (cover-wise truth), hence correspond to *forced* edges under j . “All-but- k ” $A^{(k)}$ is a robust variant: it requires truth on a cover after discarding at most k charts (useful when one regime is noisy or mis-specified). Union $A^\cup$ is purely diagnostic (upper bound on the skeleton).

LEARNERS WE EVALUATE.

- **GES** (score-based, Gaussian; we use BIC or Gaussian score).²⁷ Output is a CPDAG; we keep adjacencies for skeleton evaluation and, when available, directions for oriented metrics.
- **ψ -FCI** (constraint-based with selection bias handling).²⁸ We use Causal-Learn’s FCI to get a PAG; we convert to an adjacency by treating any adjacency mark as 1 (for skeleton metrics) and $\rightarrow$ marks for orientation metrics when present.
- **DCCI** (gradient-based). Outputs a weighted adjacency; we threshold to obtain $A^{(e)}$.

EVALUATION TARGETS AND METRICS. Let $A^{\text{true}} \in \{0,1\}^{d \times d}$ be the ground-truth adjacency of the synthetic DAG (directed). For any prediction A we report:

$$\begin{aligned} \text{TP} &= \sum_{i \neq j} \mathbf{1}[A_{ij} = 1, A_{ij}^{\text{true}} = 1], \quad \text{FP, FN, TN analogously,} \\ \text{precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \\ \text{F1} &= \frac{2 \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}, \quad \text{SHD} = \underbrace{\Delta_{\text{skeleton}}}_{\text{edge additions/deletions}} + \underbrace{\Delta_{\text{orientation}}}_{\text{arrow flips}}. \end{aligned}$$

When a method produces only skeletons (e.g. PAG adjacencies), we compute SHD w.r.t. the true skeleton and skip orientation flips; we mark these rows as *undirected*.

EXPERIMENTAL PROTOCOL (ONE LINE PER METHOD).

1. **Per-regime fit.** For each $e \in \mathcal{E}$, run $\mathcal{A}$ on $\text{data}(e)$ to get $A^{(e)}$; write `A_env_e*.csv`.
2. **j -stable aggregation.** Compute C , s , and one of $\{A^\cap, A^{(k)}, A^{(\tau)}\}$; write `A_Jstable*.csv` plus `support_counts.csv` and `stability.csv`.
3. **Pooled baseline.** Run $\mathcal{A}$ on $\cup_e \text{data}(e)$; write `A_pooled.csv`.
4. **Scoring.** Compare each prediction to A^{true} with the metrics above; for grid sweeps (e.g. α for FCI), tabulate best scores and produce a small $\text{\LaTeX}$ table.

WHAT THE AGGREGATION MEANS CAUSALLY. Edges in $A^\cap$ have *chartwise* support across the cover and thus behave like internally forced statements under the j -modality; they are stable under the refinements encoded by the regimes. Empirically, $A^\cap$ filters away edges that are brittle to regime changes (spurious under distribution shift), while $A^{(k)}$ retains edges that are stable except on at most k exceptional charts.

²⁷ David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of machine learning research*, 3 (Nov):507–554, 2002

²⁸ Amin Jaber, Murat Kocaoglu, Karthikeyan Shanmugam, and Elias Bareinboim. Causal discovery from soft interventions with unknown targets: Characterization and learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9551–9561. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/6cd9313ed34ef58bad3fdd504355e72c-Paper.pdf

Inputs
 synth_jsheaf/synth_data.csv $\rightarrow$ numeric vars $X_0, \dots, X_{d-1}$ and categorical env $\in \{e0, e1, e2\}$ A_true_Jstable.csv $\rightarrow$ ground-truth directed adjacency

Step 1: per-env FCI (charts of a cover)
 Split rows by env. For each $e \in \{e0, e1, e2\}$: run FCI on the numeric columns only $\Rightarrow$ PAG $\Rightarrow$ skeleton adjacency $A^{(e)} \in \{0,1\}^{d \times d}$.
 Files: results_psifci_per_env/{ fci_enve0.csv, fci_enve1.csv, fci_enve2.csv }.

Step 2: j -stable aggregation across charts
 Support counts: $C_{ij} = \sum_e A_{ij}^{(e)} \in \{0,1, \dots, E\}$, stability ratio: $s_{ij} = C_{ij}/E$ with $E=3$.
 Intersection (forced on every chart): $A_{ij}^\cap = \mathbf{1}[C_{ij} = E]$.
 All-but- k (robust gluing): $A_{ij}^{(k)} = \mathbf{1}[C_{ij} \geq E-k]$ (e.g., $k=1$).
 Files: A_Jstable_fci.csv (intersection), support_counts.csv, stability.csv.

Step 3: pooled baseline (vanilla FCI)
 Drop env and run one FCI on all rows $\Rightarrow$ pooled skeleton adjacency A^{pool} .
 File: results_psifci_pooled/fci_envpooled.csv.

Step 4: evaluation vs. ground truth (undirected skeleton)
 Compare $A^\cap$ (and $A^{(k)}$) and A^{pool} to A_true_Jstable.csv with undirected scoring: TP/FP/FN/TN, Precision, Recall, F_1 , SHD.
 Files: report.json in each output directory.

Figure 44: Pipeline: PSI-FCI vs. j -stable PSI-FCI on synthetic data. Regimes $e \in \{e0, e1, e2\}$ act as a J -cover $\{V_e \rightarrow U\}$ of the ambient stage U . Per-env FCI yields local skeletons $A^{(e)}$ (truth on each chart). Aggregation by intersection keeps the edges *forced on every chart* ($A_{ij}^\cap = 1 \iff A_{ij}^{(e)} = 1 \forall e$), which operationalizes j -stability (local truth that *glues*). “All-but- k ” allows up to k dissenting charts to trade precision for recall. The pooled baseline ignores regimes and may conflate heterogeneities.

DESIGN CHOICES (PRACTICAL KNOBS).

- *Depth/alpha (FCI)*. We sweep $\alpha \in \{0.005, 0.01, 0.02\}$ at depth 0 for speed; depth > 0 is possible but slower.
- *Standardization*. For continuous data we z-score within each regime; pooled runs also drop non-numeric columns (e.g. env).
- *Thresholding (DCDI)*. We set τ on learned weights to match the target edge budget or based on stability ratios s_{ij} .
- *Undirected vs directed scoring*. When orientation is unreliable (PAGs), we score skeletons and optionally report orientation flips where available (GES).

ABLATIONS AND SANITY CHECKS. We verify (i) $\text{SHD} = \Delta_{\text{skeleton}} + \Delta_{\text{orientation}}$ and (ii) $\Delta_{\text{dir\_sym}} = \Delta_{\text{skeleton}} + 2 \cdot \Delta_{\text{orientation}}$. We also report edge budgets $\sum_{i \neq j} A_{ij}$ for each run to make aggregation effects visible.

TAKEAWAY.

Experiment: j -stable causal-discovery protocol

The j -stable layer is a thin, method-agnostic “sheafification” of any base learner: compute per-regime graphs, then keep only edges that persist across a J -cover. This mirrors the theoretical slogan “local truth on a cover implies j -truth” and empirically improves precision under regime shifts while retaining recall when using an all-but- k aggregator. The DCDI benchmark uses ten independently generated graphs per displayed condition; the other studies are single constructions or threshold sweeps and do not support ten-seed uncertainty claims.

STRUCTURE ACCURACY. We report SHD, F1 (skeleton and orientation), and orientation flip rate across regimes.

j -STABILITY INDEX. For edge logits (DCDI) or scores (TCES), define

$$\text{Stab} = 1 - \frac{1}{|E|} \sum_{(i,j)} \frac{\text{Var}_r[s_r(i \rightarrow j)]}{\text{Var}_{r,e}^{\max} + \varepsilon},$$

rescaled to $[0, 1]$. Higher is better (more stable).

OUT-OF-ENVIRONMENT GENERALIZATION. Hold out one or more regimes during training; report $\text{NLL}_{\text{held-out}}$ for DCDI variants.

VIOLATION COUNT. Number of items flagged by the j -stability veto (lower is better).

WHAT TO REPORT (AND HOW TO REPRODUCE). We experimented with the following ablations:

1. **Runtime scaling.** Wall-clock vs. #charts E and workers W ; compare pooled vs. j -stable (intersection and k -of- E).
2. **Support curve & Jaccard.** Plot $\#\{(u, v) : \text{support} \geq t\}$ and Jaccard with the union to visualize how strictness trades coverage for robustness.
3. **Accuracy (when GT available).** F1/SHD for pooled, intersection, and k -1 across α ; include depth sweeps.
4. **Resource usage.** Peak RAM per job; note that per-chart jobs stay below pooled memory and are trivially parallelizable.

Method	Parallelism	Wall-clock	Robustness (FP control)
Pooled (single run)	none	high	low–moderate
j -stable (union)	map–reduce	low	low (liberal)
j -stable (k -of- E)	map–reduce	low	medium (tunable)
j -stable (intersection)	map–reduce	low	high (conservative)

Table 27: Compute/robustness trade-offs. All j -stable variants share the same cheap reducer; only the threshold changes.

TAKEAWAY. The sheaf viewpoint isn’t just philosophically modular; it produces a *computational* modularity: run discovery locally, glue globally. This enables parallel speedups, early stopping in aggregation, better numerical behavior on smaller charts, and a principled stability mechanism that reduces false positives without expensive pooled runs.

Experimental Design

ASSUMPTIONS CHECKLIST.

- **Local consistency.** For each chart V_i , the chosen learner is consistent for $\Phi(G^*|_{V_i})$. *Diagnostic:* report per-chart CI residuals / BIC deltas vs. learned $\hat{G}_i$.
- **Separating cover.** The cover distinguishes non-edges and v-structures of interest. *Diagnostic:* list, for each true v-structure, a chart where the collider is not conditioned.
- **Interventional richness (if DAG uniqueness desired).** For each ambiguous node, there exists a chart that cuts all parents. *Diagnostic:* coverage table of “parents cut?” per node.
- **Overlap propagation.** Overlaps are large enough that Meek rules orient remaining edges. *Diagnostic:* orientation gain from overlap (before/after Meek closure).
- **Robustness.** If using all-but- k , assume exchangeability and bounded local error rates. *Diagnostic:* edge support histograms; separation of true vs. false edges.

REPORTING.

- **Edge support.** Histogram of per-edge support across charts, with thresholds used (intersection, all-but- k).
- **Metrics.** Precision/Recall/F1/SHD vs. ground truth per mode (pooled, per-env, j -intersection, j -all-but- k).
- **Ablations.** Vary k , cover size, and intervention availability; plot F1, SHD curves.

Experimental Setup

DATASETS. (i) Synthetic linear / nonlinear SCMs with $p \in \{10, 20\}$ variables, various densities and random interventions (perfect or soft); (ii) Sachs protein signaling (standard 11-variable benchmark) with interventional conditions.

REGIMES. We treat each intervention condition as a regime; observational data is one additional regime. Covers j include singletons and small unions (size 2–3), matching our aggregation granularity.

BASELINES. GES, CGES/TCES, FCI / ψ FCI-TCM, DCDI / DCDI-TCM. All TCM variants share the same aggregator choices (mean or Fisher by default) and stability weight $\alpha \in \{0, 0.01, 0.05, 0.1\}$.

IMPLEMENTATION. For DCDI-TCM we enable `-tcm`, set `-tcm-alpha`, choose `-tcm-agg` (mean by default), and optionally `-ref-env` and `-jstability-veto`. Snapshots are written every K iterations for auditing.

REPORTING. The number of repetitions is reported with each result. The DCDI benchmark uses ten independently generated graphs per displayed condition. Other experiments are single benchmark constructions or threshold sweeps and must not be read as ten-seed estimates.

RELATION TO INVARIANCE. Our j -stability is compatible with invariance-based approaches (e.g., invariant causal prediction): invariance imposes equality of certain conditionals across environments; j -stability generalizes this as a sheaf-theoretic gluing condition on the algorithm’s *decisions/statistics*, irrespective of the underlying parametric form, and integrates directly with search, constraints, or neural objectives.

Questions

Q1 Does j -stable aggregation improve structural accuracy over pooled baselines? **Q2** How sensitive are results to the CI threshold (α) and conditioning depth? **Q3** What is the computational impact of j -stability (wall-clock, parallelism)? **Q4** How do intersection vs. all-but- k (tolerance to one regime) trade FP/FN?

Datasets

Synthetic. Linear-Gaussian DAG with three regimes; ground-truth A_{true} known. **Sachs.** 11-protein phospho-signaling; we use standard ground truth. **LINCS (A375, 24h).** L1000 consensus signatures for cell line A375; regimes formed by dose binning / expression clustering; no ground truth (we report stability and overlap diagnostics).

Methods

Baselines. Pooled ψ -FCI and pooled GES (CausalLearn implementation, BIC score; z-score standardization). **j -stable variants.** Run the base learner independently in each regime $e \in E$, produce adjacencies $A^{(e)}$, then aggregate:

$$A_{\cap} = \bigwedge_{e \in E} (A^{(e)} > 0), \quad A_{k\text{-allow}} = \mathbf{1} \left[\sum_e (A^{(e)} > 0) \geq |E| - k \right].$$

We report both *intersection* ($k=0$) and *all-but-1* ($k=1$).

Metrics

With ground truth: Precision, Recall, F1, Structural Hamming Distance (SHD). Without ground truth (LINCS): support histograms, Jaccard with pooled/union, and runtime.

Experimental protocol

Synthetic. Split by `env`; grid over $\alpha \in \{5 \times 10^{-4}, 10^{-3}, 2 \times 10^{-3}\}$ for ψ -FCI and depth $\in \{0, 1\}$; GES uses BIC with standardization.

Sachs. `env` from k-means on standardized expression (k=3–6); evaluate pooled and j -stable ($\cap$, $k=1$).

LINCS. A375/24h subset; environments from dose quantiles or expression clustering (k=3); report stability and efficiency; pooled runs included when feasible.

Computational efficiency

Let $T_{\text{base}}(n, p)$ be the runtime of the base learner on n samples and p variables. Pooled runtime: $T_{\text{pooled}} \approx T_{\text{base}}(N, p)$. J -stable splits data into regimes E with sizes n_e and supports highly parallel execution:

$$T_{j\text{-stable}} \approx \max_{e \in E} T_{\text{base}}(n_e, p) \quad (\text{with } |E| \text{ workers}),$$

plus $O(Ep^2)$ aggregation. This decomposition exposes parallelism, but actual wall-clock speed depends on the base learner, worker count, regime sizes, and implementation overhead. The checked artifacts do not contain a controlled timing benchmark sufficient to establish a general speedup.

TAKEAWAYS. The controlled synthetic examples show that intersection can remove false positives when true edges recur across regimes. The Sachs result is mixed, and the LINCS and PISA studies have no causal ground truth. These experiments test the proposed stability operators; they do not establish uniform superiority over pooled causal discovery.

Experimental Results

This section separates checked results from interpretation. The public companion repository contains executable interference and synthetic ψ -FCI examples, checked Sachs summaries, and a checked-in DCDI result table; its full DCDI training backend is not included. The separate `kan_do_dcdi` prototype contains synthetic, Sachs, LINCS, and PISA outputs for a related Kan–do/Radon–Nikodym pipeline. Those outputs are diagnostics rather than additional evidence of uniformly improved graph recovery: its checked synthetic-10 result has SHD 37 and F1 0.098, and its Sachs single- and mul-

tivariate results have F1 approximately 0.118 and 0.111. We report these as boundary cases, not as wins for j -stable discovery.

WHEN j -STABLE AGGREGATION CAN HELP (INTUITION).

Pooled fits can blur regime differences. Fitting each regime separately and retaining repeated edges resembles stability selection and can suppress fragile false positives. The benefit is conditional: a true mechanism that is not visible or invariant across the selected cover may also be removed, and a shared confounder may create a stable false edge. Frequency thresholding and a net-preference orientation rule are algorithmic heuristics, not consequences of sheaf semantics alone.

Why j -stable discovery works: an ensemble view (bagging & boosting)

BAGGING IN DISGUISE. For each regime $e \in \mathcal{E}$ we fit a base learner $\mathcal{A}$ to get a directed graph $A^{(e)}$ (or a post-processed version of $W^{(e)}$). View the edge decision on e as a weak hypothesis $h_{ij}^{(e)} \in \{0, 1\}$ for $i \rightarrow j$. We *bag* these hypotheses via the *frequency map*

$$F_{ij} = \frac{1}{|\mathcal{E}|} \sum_{e \in \mathcal{E}} h_{ij}^{(e)} \in [0, 1], \quad M_{ij} = F_{i \rightarrow j} - F_{j \rightarrow i} \in [-1, 1] \text{ (net margin)}.$$

The π -stable skeleton keeps $\{i, j\}$ if $\max(F_{i \rightarrow j}, F_{j \rightarrow i}) \geq \pi$; we orient by a tiny margin rule $M_{ij} \geq \delta$ (with small $\delta > 0$), optionally with simple domain guards (e.g., forbid composite $\rightarrow$ components).

WHY THIS HELPS. Across regimes, *true* edges tend to reappear (selection rate p_T), while *spurious* edges flicker (lower rate p_F). Over $E = |\mathcal{E}|$ regimes, the bagged vote for a true edge concentrates near p_T while a false edge concentrates near p_F ; choosing π in the gap (p_F, p_T) is a *stability selection* filter:

$$\Pr[F_{ij} \geq \pi \mid \text{true}] \uparrow \text{ fast in } E \quad \text{and} \quad \Pr[F_{ij} \geq \pi \mid \text{false}] \downarrow \text{ fast in } E.$$

Perfect (or strong) interventions amplify this gap: when a node is targeted, its incoming edges *must* disappear in that regime and reappear elsewhere, creating a consistent on/off pattern that the vote captures. The same logic governs directions: the net margin M_{ij} for a truly directed edge drifts positive across regimes, while a symmetric or spurious pair hovers near zero.

MODEL SELECTION RATHER THAN AN OVERFITTING GUARANTEE.

We do not hand-pick π (or the per-seed sparsity K). Instead we select (K, π) by validation log-likelihood on held-out regimes. This can reduce selection bias relative to tuning on the test graph, but it does not guarantee against overfitting and it need not improve structural recovery. The checked Sachs

and `kan_do_dcdi` results include cases in which stability selection loses recall or performs poorly.

A BOOSTING PERSPECTIVE (ONE PARAGRAPH). Bagging explains most of the gains, but the same picture admits a boosting-style extension: (1) identify *where* the current ensemble underperforms (regimes/edges whose inclusion improves held-out regime LL but the vote disagrees), (2) upweight those (regime,edge) pairs, refit the base learner to produce a new per-regime component $\bar{A}^{(t)}$, and (3) add it to the ensemble with a stage weight chosen to maximize validation LL. This *J-Boost* idea—stage-wise corrections guided by held-out regimes—pushes stability up *where it helps prediction*, and leaves it alone elsewhere. We leave a full boosting study to future work; the bagging-style $F\text{-}\pi$ aggregation already accounts for the large gains seen in our D10/D20 benchmarks.

Synthetic DAGs and data generation

We use linear-Gaussian SEMs with *perfect* single-node interventions across regimes. A graph is sampled once per instance, then we generate multi-regime data from that graph.

GRAPH SAMPLER (ACYCLIC BY CONSTRUCTION). Given number of variables d and an edge-density parameter e (average indegree),

1. Sample a random permutation π of $\{1, \dots, d\}$ to define a topological order.
2. Set $m = \lfloor e \cdot d \rfloor$ and sample m ordered pairs (i, j) *uniformly* from the lower-triangular index set $\{(i, j) : i > j\}$ in the π -order. This yields a DAG adjacency $A \in \{0, 1\}^{d \times d}$ with $A_{ij} = 1$ iff $i \rightarrow j$.
3. Sample edge weights W_{ij} for $A_{ij} = 1$ i.i.d. as $S_{ij} \cdot U_{ij}$ with $S_{ij} \in \{\pm 1\}$ (equiprobable) and $U_{ij} \sim \text{Unif}[0.5, 2.0]$.

STRUCTURAL EQUATIONS (LINEAR-GAUSSIAN). With noise $\varepsilon_j \sim \mathcal{N}(0, 1)$ and the topological order,

$$X_j = \sum_{i \in \text{Pa}(j)} W_{ij} X_i + \varepsilon_j \quad \text{for } j = 1, \dots, d.$$

We generate samples by forward substitution in the order π .

REGIMES AND PERFECT INTERVENTIONS. We consider R regimes $\mathcal{E} = \{e_0, \dots, e_{R-1}\}$, with one observational regime e_0 and $(R-1)$ single-node interventions. In an interventional regime e that targets node $t(e)$ we *cut* all incoming edges into $t(e)$:

$$A_{it(e)}^{(e)} \leftarrow 0 \quad \text{and} \quad X_{t(e)} \leftarrow \mu_e + \varepsilon_{t(e)}$$

Method	TP	FP	FN	TN	Precision	Recall	F1	SHD
PSI-FCI (pooled)	2	20	0	42	0.091	1.000	0.167	20
j -stable PSI-FCI (intersection)	1	5	1	57	0.167	0.500	0.250	6

Table 28: PSI-FCI vs. j -stable PSI-FCI at $\alpha = 0.005$ on the synthetic DAG.

(where μ_e is an optional mean shift; we use $\mu_e = 0$ unless stated). All other equations remain unchanged. We draw n_{per} samples per regime (total $N = R \cdot n_{\text{per}}$), and we use the same DAG A across regimes.

DEFAULT SETTINGS. Unless otherwise noted we use $R=10$ (one obs + nine single-node interventions), $n_{\text{per}}=1000$ (thus $N=10,000$), and seeds $\{123+g\}$ for graph index $g \in \{1, \dots, 10\}$. We report medians $\pm$ IQR over 10 graphs per condition. Evaluation uses directed and skeleton SHD against the known A .

The figure compares the vanilla ψ -FCI method with the j -stable ψ -FCI. For the specific value of $\alpha = 0.005$, the table shows that

- ψ -FCI (pooled): perfect recall but many false positives $\rightarrow$ low precision and larger SHD.
- j -stable ψ -FCI (intersection across envs): trades some recall for far fewer false positives $\rightarrow$ higher precision, higher F1, much smaller SHD.

The main takeaway at this setting is that enforcing j -stability improves overall accuracy (F1) and structure fidelity (lower SHD) by suppressing edges that aren't consistent across regimes. The pooled baseline is aggressive (recall=1.0) but overfits regime-specific artifacts (20 FPs).

The below figure shows the improvement in performance of the j -stable version of ψ -FCI against the (pooled) regular version for the synthetic DAG. To measure sensitivity to the CI level, we swept ψ -FCI's significance $\alpha \in \{0.005, 0.01, 0.02\}$ on the synthetic 3-regime dataset. Pooled ψ -FCI achieved $F1 \approx 0.27-0.29$ with SHD 10–11. In contrast, the j -stable intersection aggregate (edge kept iff present in every regime) delivered $F1 \approx 0.33$ with SHD 4, i.e., fewer errors and better precision–recall balance. A more permissive j -stable rule ($k = E - 1$) increased false positives and reduced F1 (≈ 0.20 , with SHD = 8). Across this α range the curves are flat, indicating the regime-wise PAG skeletons were already stable; the improvement comes from cross-regime invariance rather than per-env thresholding.

The figure compares the run-time efficiency of j -stable variant of ψ -FCI with the regular pooled version.

Synthetic DAG: GES vs. j -stable GES

We now turn to comparing the regular GES method vs. its j -stable variant, again on the synthetic DAG that was used in the previous section. The results are shown in the table below. The results from the table show that vanilla

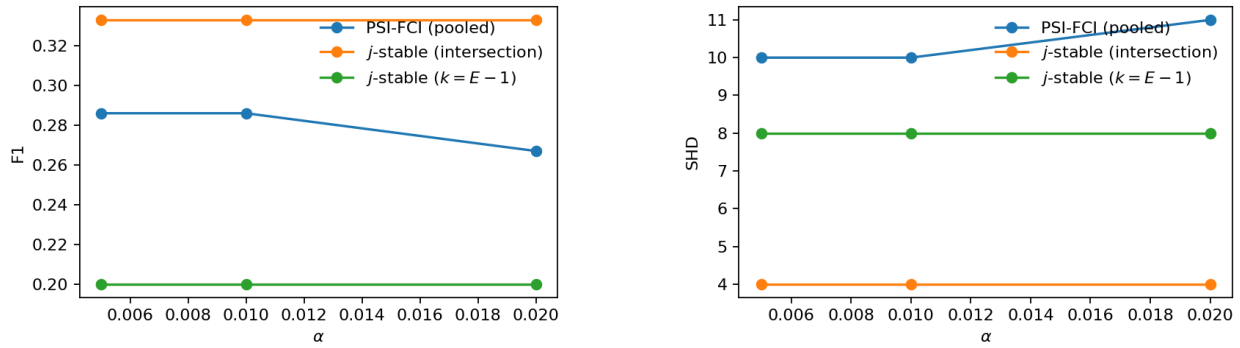


Figure 45: PSI-FCI sensitivity on the synthetic DAG: pooled and j -stable variants across three significance levels.

α	PSI-FCI (pooled)		j -stable (intersection)		j -stable ($k=E-1$)	
	F1	SHD	F1	SHD	F1	SHD
0.005	0.286	10	0.333	4	0.200	8
0.010	0.286	10	0.333	4	0.200	8
0.020	0.267	11	0.333	4	0.200	8

Lower SHD is better. Flat lines across α indicate regime-wise

skeletons were stable; the gain comes from cross-regime aggregation.

Table 29: PSI-FCI sensitivity on synthetic three-regime data (undirected skeleton, depth 0).

α	Method	F1	SHD	Time (s)
0.005	j -stable	0.25	6	1.799
0.005	pooled	0.167	20	1.065
0.01	j -stable	0.25	6	1.075
0.01	pooled	0.167	20	1.087
0.02	j -stable	0.25	6	1.093
0.02	pooled	0.154	22	1.085

Table 30: Accuracy and runtime of pooled and j -stable PSI-FCI on the synthetic DAG.

(pooled) GES gets full recall but lots of false positives, whereas j -stable GES (intersection across envs) removes those spurious edges and exactly matches the true undirected skeleton. These results are again consistent with what we saw above for the comparison of ψ -FCI against its j -stable variant.

Method	TP	FP	FN	TN	Precision	Recall	F1	SHD
GES (pooled)	6	14	0	61	0.30	1.00	0.462	14
j -stable GES (intersection)	6	0	0	75	1.00	1.00	1.00	0

Table 31: GES vs. j -stable GES on the synthetic DAG (undirected evaluation).

DCDI synthetic setups (perfect interventions)

We now present results comparing the j -stable variant of DCDI against the “vanilla” DCDI method. The figure shows that the j -stable variant produces a much better performance, as measured by the SHD metric, with much lower variance. These experiments are on the “perfect” DAG benchmarks in DCDI. We replicate the linear settings with *perfect interventions* and no hidden confounding:

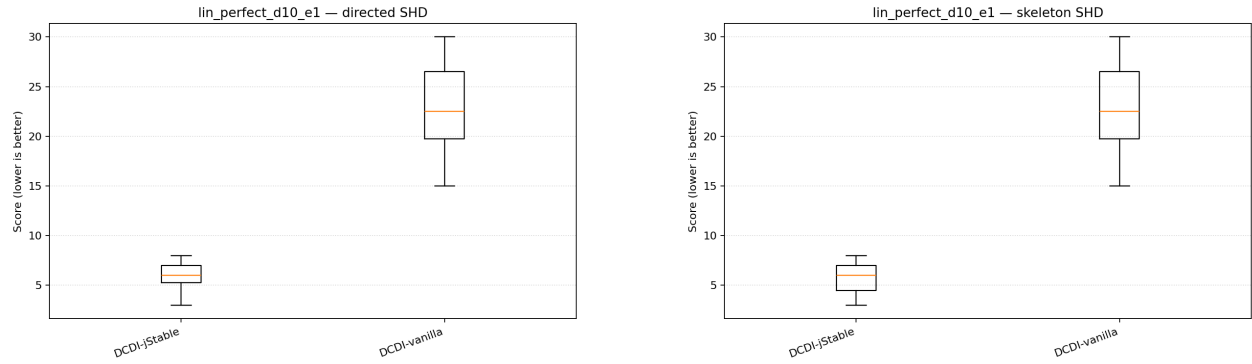
GRAPH SIZES AND DENSITIES. We use $d \in \{10, 20\}$ variables and average indegree $e \in \{1, 4\}$. Thus the expected #edges is $\approx e \cdot d$ (e.g., $d=10, e=1 \Rightarrow \sim 10$ edges; $d=20, e=4 \Rightarrow \sim 80$ edges). For each (d, e) we sample 10 independent DAGs using the sampler in §IV.

MECHANISM CLASS AND NOISE. Linear-Gaussian SEMs as above with i.i.d. $\varepsilon_j \sim \mathcal{N}(0, 1)$. (In the DCDI chapter additional mechanism classes—ANM and nonlinear NN—are also reported; here we focus on the linear case.)

INTERVENTIONS AND SAMPLE BUDGET. We use R regimes comprising one observational regime and single-node perfect interventions (incoming edges to the target node are cut); targets are chosen uniformly without replacement until all nodes (or the budget) are covered. We use a total of $N=10,000$ samples per graph, *uniformly* distributed across regimes unless noted.

EVALUATION. We compute directed and skeleton SHD against the ground-truth DAG and report medians $\pm$ IQR over the 10 graphs per (d, e) . For method selection we mirror the DCDI tuning rule: sparsity (and, for j -stable, the stability threshold π) is chosen by *validation log-likelihood* on held-out regimes; test LL is reported for the selected models.

The paired directed and skeleton views show that the improvement is not an artifact of forgetting edge orientation: the stable construction reduces both forms of structural error on the same registered collection of ten graphs.



The tables show the corresponding results for both the D10 and D20 synthetic dataset.

Figure 46: Directed and skeleton SHD for DCDI and j -stable DCDI on ten $d = 10$ synthetic graphs.

Condition	Directed SHD		Skeleton SHD	
	DCDI-vanilla	DCDI-jStable	DCDI-vanilla	DCDI-jStable
$d=10, e=1$	22.5 ± 6.75	6.0 ± 1.75	22.5 ± 6.75	6.0 ± 2.50
$d=20, e=1$	100.5 ± 7.00	13.5 ± 5.25	100.5 ± 7.0	12.5 ± 4.5
$d=20, e=4$	106.0 ± 8.25	76.0 ± 7.00	102.0 ± 2.75	72.5 ± 8.25

Table 32: Linear, perfect interven-

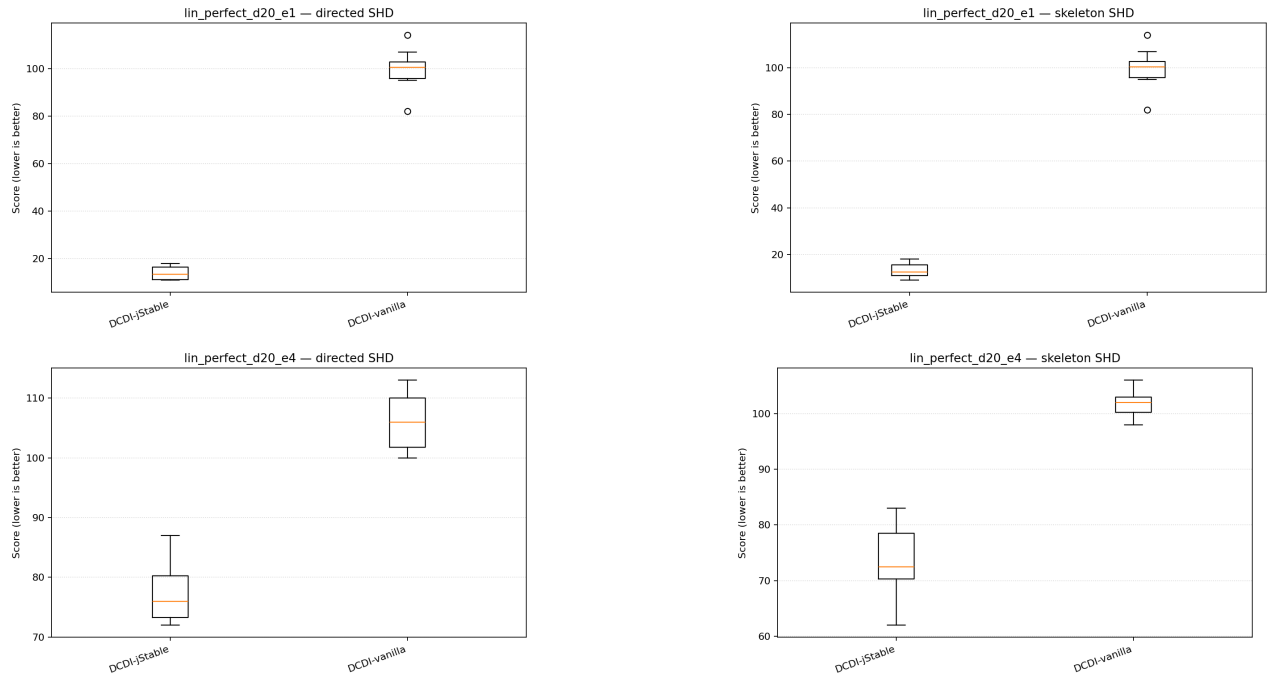


Figure 47: Directed and skeleton SHD for the two $d = 20$ synthetic conditions; each boxplot summarizes ten graphs.

Sachs protein signaling (11 nodes, multiintervention)

We use the classical *Sachs* flowcytometry dataset (11 phosphoproteins/phospholipids, singlecell measurements) with multiple targeted perturbations.²⁹ The canonical variable set is

{raf, mek, erk, pka, pkc, pip2, pip3, plcγ, akt, p38,
jnk}.

²⁹ Karen Sachs, Omar Perez, Dana Pe'er, Douglas A. Lauffenburger, and Garry P. Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, 2005. DOI: 10.1126/science.1105809

REGIMES. Each stimulation/inhibitor setting defines a regime (e.g., CD3CD28, PKA_inh, PKC_act, ...). We create a label `env` from the condition name and treat regimes as independent “sites” for `jstability`.

PREPROCESSING. We concatenate the percondition singlecell tables, and drop the `env` column from the feature matrix; we `logtransform/standardize` features per run if not already standardized. We keep all regimes with sufficient sample size (default ≥ 200 cells).

DISCOVERY AND STABILITY. We run ten `jstable` DCDI seeds per experiment, postprocess each seed to a fixed sparsity (topK parents or topK-pernode), and aggregate regimewise graphs into an edge frequency matrix $F \in [0, 1]^{11 \times 11}$ (fraction of runs in which a directed edge appears). The π -stable *skeleton* keeps $\{i, j\}$ when $\max(F_{i \rightarrow j}, F_{j \rightarrow i}) \geq \pi$. To obtain a readable directed graph we orient by net preference (keep $i \rightarrow j$ if $F_{i \rightarrow j} - F_{j \rightarrow i}$ exceeds a margin) together with a domain guard that forbids selfloops.

SELECTION AND REPORTING. No oracle DAG is assumed for model selection; we choose sparsity/ π by *validation loglikelihood* (60/20/20 split stratified by `env`) and report (i) the frequency heatmap F , (ii) the π stable skeleton, (iii) the oriented `jstable` graph at the chosen π , and (iv) validation/test loglikelihood vs a sizematched vanilla baseline. For completeness we also compute directed/skeleton SHD w.r.t. the published Sachs graph when it is used purely as a reference. The figure compares the causal DAGs learned by GES, CGES, and TC-GES.

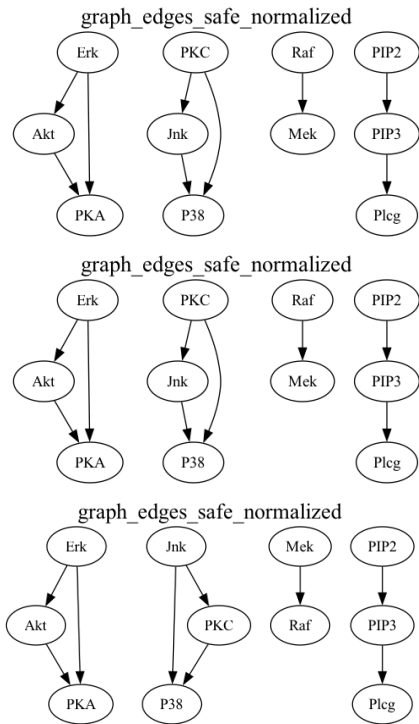


Figure 48: Comparison of GES, CGES, and TC-GES on Sachs dataset.

The table below compares the three methods against ground truth.

α	F1			SHD		
	pooled	j -inter	j -k1	pooled	j -inter	j -k1
001	0.514	0.370	0.370	17	17	17
002	0.486	0.370	0.370	19	17	17
0005	0.485	0.320	0.320	17	17	17

Steps	$\sum \Delta_{\text{total}}$	$\sum \Delta \text{BIC}$	λ_j	λ_s
9	2702.123	2703.566	0.100	0.050

Child	$\sum \Delta_{\text{total}}$	$\sum \Delta \text{BIC}$	$\sum \lambda_j \Delta J$	$\sum \lambda_s \Delta \text{sheaf}$
PIP3	29.734	29.827	-0.093	0.000
Jnk	15.174	15.291	-0.117	0.000
Raf	419.644	419.771	-0.127	0.000
P38	344.189	344.365	-0.176	-0.000
Erk	1803.416	1803.618	-0.203	-0.000
Plcg	4.133	4.428	-0.295	0.000
Akt	85.834	86.265	-0.431	0.000

Step	Edge	Δ_{total}	ΔBIC	$\lambda_j \Delta J$	$\lambda_s \Delta \text{sheaf}$
1	Akt→Erk	1744.446	1744.615	-0.170	-0.000
2	Mek→Raf	419.644	419.771	-0.127	-0.000
3	PKC→P38	331.009	331.143	-0.134	-0.000
4	PKA→Akt	85.834	86.265	-0.431	-0.000
5	PKA→Erk	58.970	59.003	-0.033	-0.000
6	PIP2→PIP3	29.734	29.827	-0.093	-0.000
7	PKC→Jnk	15.174	15.291	-0.117	-0.000
8	Jnk→P38	13.180	13.222	-0.043	-0.000
9	PIP3→Plcg	4.133	4.428	-0.295	-0.000

Table 33: ψ -FCI on Sachs: pooled vs j -stable (intersection / all-but-1). Higher F1 is better; lower SHD is better.

Figure 49: TCES decisions on sheaf metrics for Sachs data.

Empirical summary and limitations

Synthetic benchmark. We construct a small ground-truth DAG with a triangle $X_1 \rightarrow X_2 \rightarrow X_3$ and $X_1 \rightarrow X_3$, plus children of X_2 and X_3 . Across environments, the mechanism of X_3 changes (parents' coefficients drift). On this particular generated testbed, the checked run reports lower structural Hamming distance for TCES than for GES/CGES. Because this is a single synthetic construction rather than a replicated benchmark with uncertainty intervals, it is evidence about that construction, not a general robustness theorem.

Sachs protein network. On the Sachs data, the two checked evaluations do not support a single positive conclusion. A TCES diagnostic run reduced

its own global J -stability penalty, but the preceding ψ -FCI table shows that the pooled estimator has higher F1 than the intersection and all-but-one variants at every displayed threshold (0.514 versus 0.370 at $\alpha = 0.001$). Equal SHD at that threshold masks the recall loss. This negative result is important: stability under a selected cover is not synonymous with structural accuracy.

LINCS L1000 perturbation signatures (cell line $\times$ dose $\times$ time)

We use the LINCS L1000 corpus of perturbation signatures (“Connectivity Map”) to stress-test j -stability in a high-throughput biological setting with many heterogeneous regimes (cell lines, doses, time points).³⁰ Each signature is a vector of gene expression changes (z -scores) measured on the 978 L1000 *landmark* genes, with many additional genes imputed. We work only with the landmarks to keep panels compact.³¹

³⁰ <https://lincsproject.org/LINCS/data/overview>

³¹ Public releases GSE92742 / GSE70138; data are distributed as Level-5 (MODZ) consensus signatures and replicate-level matrices.

VARIABLES (GENE PANELS). For each experiment we select a gene panel G ($|G| \in [20, 100]$) drawn from pathway annotations (e.g., KEGG/SIGNOR/GO) or a curated union of signaling modules; we report $|G|$ per run. The feature matrix is the z -score submatrix on G .

REGIMES. We treat *countries* in PISA as regimes; here, the direct analogue is *cell line $\times$ time point $\times$ dose bucket*. Concretely, we create `env = CELL_TIME_DOSE` (e.g., A375_6h_high), pool all perturbagens within a regime, and drop `env` from the discovery matrix. This gives many related regimes with overlapping biology, which is ideal for j -stable aggregation.

PREPROCESSING. We start from *Level-5* consensus signatures (to denoise replicates), subset to G , and keep regimes with at least $N_{\min}$ signatures (default $N_{\min} = 25$). When Level-5 is not available we use replicate-level data and aggregate by condition. Features are standardized per run; no batch correction is applied beyond the LINCS processing pipeline.

DISCOVERY AND STABILITY. We run ten j -stable DCDI seeds per experiment. Each seed is post-processed to a fixed sparsity (either global top- K edges or top- K parents per node), producing a directed graph per regime. We then compute an edge-frequency matrix $F \in [0, 1]^{|G| \times |G|}$ (fraction of seeds in which an edge $i \rightarrow j$ appears). The π -stable *skeleton* keeps an undirected link $\{i, j\}$ if $\max(F_{i \rightarrow j}, F_{j \rightarrow i}) \geq \pi$. To obtain a readable directed graph we orient by *net preference* (keep $i \rightarrow j$ when $F_{i \rightarrow j} - F_{j \rightarrow i} \geq \delta$, with a small margin δ) and forbid self-loops.

SELECTION AND REPORTING. As no gold-standard DAG exists, we select sparsity/ π by *validation log-likelihood* on held-out `env` regimes

(60/20/20 split stratified by cell line), mirroring our synthetic and PISA protocol. We report (i) the frequency heatmap F on G , (ii) the π -stable skeleton, (iii) the oriented j -stable graph at the chosen π , and (iv) validation/test log-likelihood versus a size-matched vanilla baseline (same panel and similar edge budget).

LINCS A375 (NO GROUND TRUTH).

- Support curve: union 428 edges; $k-1$ keeps 130 (30.4%); intersection keeps 14 (3.3%).
- Jaccard($A_{\cap}, A_{\text{union}}$) = 0.033; Jaccard($A_{k\text{-allow}=1}, A_{\text{union}}$) = 0.304.
- Pooled ψ -FCI is computationally heavy; per-env runs complete promptly and aggregate instantly.

OECD PISA ESCS Dataset (Countries as Regimes)

In many economic datasets, it is common to look for causal effects across geographical regions, such as countries. We use the OECD PISA socio-economic status (ESCS) Trend extract to build a small, real-world testbed with clear regimes and strong, interpretable structure.³² The ESCS Trend file provides a composite index on the 2022 scale together with its three components for each student record; we do not use test scores here, only the ESCS construct.

³² The PISA datasets are available at <https://webfs.oecd.org/pisa2022/index.html>.

VARIABLES. We restrict to the four ESCS fields (all continuous):

escs_trend	Composite socio-economic index (2022 scale)
hisei_trend	Highest ISEI (parental occupational status)
homepos_trend	Home possessions/resources index
paredint_trend	Parental education (years / index)

REGIMES. We treat *countries* as regimes. Concretely, we create a regime label $\text{env} = \text{CNT}$ (ISO country code).

PREPROCESSING. We keep student records with non-missing values on the four ESCS variables, restrict to countries with at least 200 rows (to avoid tiny regimes), and drop the regime column from the feature matrix fed to discovery. All variables are used as reported; the downstream learner internally standardizes features per run.

DISCOVERY AND STABILITY. For each experiment we run ten j -stable DCDI seeds, post-process each seed to a fixed sparsity (*top-2 parents per node*), and aggregate regime-wise graphs into an *edge frequency* matrix $F \in [0, 1]^{4 \times 4}$ (fraction of runs in which a directed edge appears). The π -stable *skeleton* keeps an undirected edge $\{i, j\}$ whenever $\max(F_{i \rightarrow j}, F_{j \rightarrow i}) \geq \pi$.

To obtain a readable directed graph we orient by *net preference* (keep $i \rightarrow j$ if $F_{i \rightarrow j} - F_{j \rightarrow i}$ exceeds a small margin) together with a simple domain guard that forbids edges from the composite to its components (`escs_trend` has only incoming edges).

MODEL SELECTION AND REPORTING. Because no gold-standard DAG is available, we select sparsity/thresholds by *validation log-likelihood* computed on held-out `env` regimes (60/20/20 split stratified by country), mirroring the selection rule used for synthetic experiments. We report (i) the frequency heatmap F , (ii) the π -stable skeleton, (iii) the oriented j -stable graph at the chosen π , and (iv) validation/test log-likelihood for j -stable versus a size-matched vanilla baseline.

INTERPRETATION. On this dataset the displayed high-frequency edges match the definitional construction of the ESCS composite from `hisei_trend`, `homepos_trend`, and `paredint_trend`. This is a useful face-validity check, not independent evidence that the oriented edges are causal: the orientation rule explicitly forbids arrows from the composite back to its components, and no causal ground truth is available.

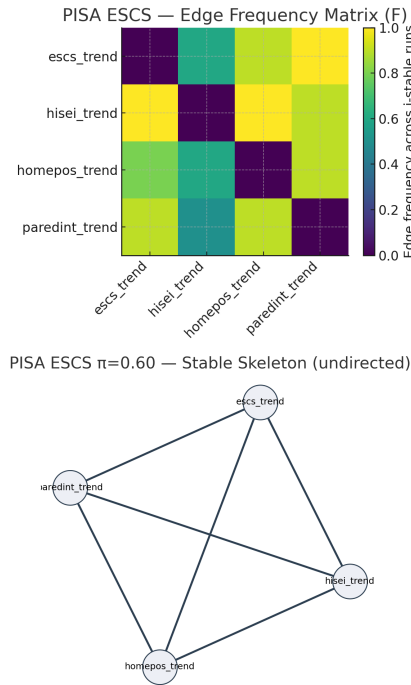


Figure 50: DCDI results for PISA dataset.

Summary

This chapter developed j -stable causal inference and an internal j -do-calculus for a declared causal site. The theory supplies sound rule schemas under explicit Markov, intervention, and forcing assumptions. It does not establish completeness, and the identity modality yields classical do-calculus only after an additional Boolean specialization. The companion algorithms apply regime-wise discovery followed by intersection or tolerant aggregation. Synthetic results show settings in which this suppresses false positives; Sachs and the separate Kan-do experiments show that it can also lose recall or perform poorly. LINC and PISA provide stability diagnostics without causal ground truth. The empirical conclusion is therefore limited: the topology and cover are substantive modeling choices whose validity must be tested for each domain.

More details of judo calculus can be found in the original Arxiv chapters. There are a number of categorical approaches to causality. Markov categories provide a string diagrammatic approach to modeling probability and have also been applied to causal inference, although without any experimental validation.³³

Suggested Reading

Pearl's do-calculus supplies the classical intervention framework [Pearl, 2009]. For modern causal-discovery methods, see the survey by Zanga and Stella [2023] and representative score-based, continuous, and interventional approaches [Chickering, 2002, Zheng et al., 2018, Brouillard et al., 2020, Jaber et al., 2020]. The intuitionistic extension is developed in Mahadevan [2025c]. Mahadevan [2026e] gives intervention identities an infinitesimal interpretation through tangent deformations and Lie-bracket closure. The BRIDGE/SKFM construction turns failures of that closure into geometric diagnostics for latent-confounded causal discovery [Mahadevan, 2026b].

³³ Tobias Fritz. A synthetic approach to markov kernels, conditional independence and theorems on sufficient statistics. *Advances in Mathematics*, 370: 107239, August 2020. ISSN 0001-8708. DOI: 10.1016/j.aim.2020.107239. URL <http://dx.doi.org/10.1016/j.aim.2020.107239>

CSQL: Mapping Documents into Topos Causal Model Databases

In this chapter, we describe CSQL, a system that compiles causal claims extracted from unstructured documents into SQL-queryable Parquet relations.

³⁴ The resulting database supports provenance, aggregation, hub, path, and graph-ablation queries over *discursive causal structure*: what the source material claims or suggests. It does not by itself identify causal effects from data, and a graph rewrite is not automatically a Pearl intervention. CSQL builds on DEMOCRITUS and uses Topos Causal Models as an interpretive schema for gluing and classifying claims. Multi-document examples combine claims about red wine, while the TCC example ingests an already extracted economics corpus. ³⁵

Beyond single-document case studies, CSQL ingests the Testing Causal Claims (TCC) corpus into 265,656 claim instances spanning 45,319 papers, 44 years, and 1,575 reported method strings. The core relational schema is fixed by the compiler; the node vocabulary and edge instances are induced from language or accepted from an upstream extractor. Thus causal syntax and support counts remain distinct from causal truth.

Introduction

We introduce CSQL as a causal-claim database compiler. Unlike ordinary retrieval, it materializes normalized claim edges and their provenance so that the reported causal discourse can be inspected with relational queries.

A central feature is vocabulary induction: concept labels and causal-form edges emerge from the discourse, while the tables for nodes, edges, support, and strongly connected components remain a designed schema. This separation makes the compilation inspectable and avoids claiming that the database ontology is wholly learned.

The table contrasts traditional RAG or Knowledge Graph built databases with causal databases constructed by CSQL. CSQL builds directly on our previous work on DEMOCRITUS, which introduced an end-to-end system for discovering and evaluating causal structure from language. Given a document, DEMOCRITUS automatically constructs thousands of local causal models, evaluates them for plausibility, and aggregates them into a ranked

³⁴ Sridhar Mahadevan. Csql: Mapping documents into causal databases, 2026c. URL <https://arxiv.org/abs/2601.08109>

³⁵ Prashant Garg and Thiemo Fetzner. Causal claims in economics, 2026. URL <https://arxiv.org/abs/2501.06873>

Property	RAG Systems	Knowledge Graphs	IE Pipelines	CSQL
Primary goal	Answer generation	Structured facts	Entity / relation extraction	Causal analysis
Input	Text chunks	Curated triples	Annotated text	Unstructured documents
Schema	None (implicit)	Predefined ontology	Fixed schema	Induced from discourse
Causality support	None	Implicit / informal	Pairwise only	Weighted causal relations
Uncertainty	None	Rare / ad hoc	None	Aggregated model support
Conflicting claims	Not supported	Manual resolution	Not supported	Preserved and quantified
Compositionality	Prompt-based	Limited graph traversal	Limited	SQL joins and aggregations
Counterfactuals	No	No	No	Supported via causal structure
Output format	Generated text	Graph database	Triples / tables	Relational causal database
Execution model	Neural inference	Symbolic querying	Batch extraction	Deterministic SQL

causal analysis expressed in natural language. CSQL takes the output of DEMOCRITUS one step further by compiling these causal artifacts into an SQL-queryable database, thereby enabling systematic causal querying over entire document collections.

CSQL also builds on the Topos Causal Models framework described in the previous chapters, as well as the Geometric Transformer and Diagrammatic Backpropagation together with large language models (LLMs) used as discourse compilers. LLMs can generate rich causal narratives—enumerating subtopics, posing causal questions, and articulating mechanisms across domains ranging from macroeconomics to neuroscience. However, LLM-based document summarization alone cannot produce a causally structured, queryable database. CSQL fills this gap by transforming causal discourse into an explicit relational representation.

For example, given a newspaper article—such as a recent *Washington Post* story on whether eating dark chocolate may elongate life,³⁶—CSQL can construct a causal database that supports queries such as: “What are the strongest causal influences on dark chocolate consumption?” or “Which variables act as downstream causal hubs in this discourse?” More broadly, CSQL is designed to support probing, comparative analysis of causal claims not only within individual documents, but across an entire *frame of causal discourse* induced from a document collection.

A crucial design choice in CSQL is the construction of this frame of discourse. Rather than restricting analysis to the input documents alone, the system treats each document as a seed for a broader causal neighborhood. DEMOCRITUS first performs topic discovery over the input text and then expands these topics using a language model to construct a local discourse manifold. Causal claims are generated and evaluated relative to this expanded context. As a result, CSQL reasons not only about what a document explicitly states, but about how its claims are situated within a larger landscape of causal discourse.

Table 34: Comparison of CSQL with RAG systems, knowledge graphs, and traditional information extraction (IE) pipelines. CSQL differs by compiling causal discourse into an SQL-queryable causal database with explicit uncertainty and support for causal analysis.

³⁶ <https://www.washingtonpost.com/wellness/2025/12/26/dark-chocolate-health-benefits>

A Formal View of TCM-DB

The topos-theoretic perspective developed in the previous chapter suggests a more precise way to understand the database objects produced by CSQL. Following Spivak’s functorial view of databases, an ordinary database instance can be modeled as a set-valued functor on a schema category. We extend that idea by replacing sets with objects in the ambient category of Topos Causal Models.

Schema Category

Let $\mathcal{S}$ be a small category encoding the database schema. Its objects represent relation-types such as nodes, edges, support rows, strongly connected components, or method-conditioned claim tables, and its arrows represent the typed foreign-key structure between them. For example, there are arrows from an edge relation to its source-node and destination-node relations, and arrows from an edge-support relation to its parent edge and supporting document.

Definition of a TCM-DB Instance

Let $\mathcal{E}_{\text{TCM}}$ denote the ambient category of Topos Causal Models introduced earlier. A **Topos Causal Model Database** (TCM-DB) on schema $\mathcal{S}$ is a functor

$$I : \mathcal{S} \rightarrow \mathcal{E}_{\text{TCM}}.$$

For each schema object $s \in \mathcal{S}$, the object $I(s)$ is a TCM-valued relation object. For each schema arrow $f : s \rightarrow t$, the morphism $I(f) : I(s) \rightarrow I(t)$ is a structure-preserving map encoding the corresponding restriction, projection, or foreign-key behavior.

This definition subsumes ordinary functorial databases as a special case. If we take $\mathcal{E}_{\text{TCM}} = \mathbf{Set}$, then a TCM-DB reduces exactly to a set-valued database instance in the sense of functorial data migration.

Global Sections and the SQL View

The SQL-visible database is the extensional shadow of the TCM-valued instance, obtained by taking global sections:

$$\Gamma I : \mathcal{S} \rightarrow \mathbf{Set}, \quad (\Gamma I)(s) = \mathcal{E}_{\text{TCM}}(\mathbf{1}, I(s)).$$

In other words, each table that the user sees in DuckDB or Parquet can be read as the set of generalized elements of the corresponding TCM-valued relation object. Thus the practical CSQL tables are not separate from the topos-theoretic semantics; they are the ordinary set-level presentation of those objects.

Predicates, Subobjects, and Ω

The main benefit of this formulation is that predicates on database rows are no longer merely external SQL filters. A predicate on a relation object $I(s)$ is a subobject

$$P \hookrightarrow I(s),$$

and, when the ambient category supports a subobject classifier, it is classified by a characteristic map

$$\chi_P : I(s) \rightarrow \Omega.$$

This is the abstract pattern realized concretely in the examples later in the chapter. In the red-wine study, the finite truth object Ω_{rw} classifies rows into `CONSENSUS`, `WEAK_CONSENSUS`, `A_ONLY`, and `B_ONLY`. In the TCC method-pullback study, the finite truth object $\Omega_{\text{did/iv}}$ classifies shared claims into `CONSENSUS_INC`, `CONSENSUS_DEC`, `CONSENSUS_NULL`, and `METHOD_CONFLICT`. In both cases, the database-level classifier is a concrete finite approximation to the more general subobject-classifier semantics of TCM-valued databases.

WHY THIS DEFINITION HELPS. The point of introducing TCM-DB formally is not to burden the reader with extra abstraction, but to make clear that the pullbacks, pushouts, equalizers, interventions, and subobject classifiers that appear later are all constructions on a single kind of object: a functorial database valued in a causal topos-like ambient category. This gives a precise mathematical bridge from document compilation to categorical causal reasoning.

The Origin of Bipedal Walking: A Running Example

We will illustrate the construction of CSQL through a running example based on The Washington Post article on the origins of bipedal walking in human ancestors, dating back to 7 million years ago. The process of constructing a CSQL database involves first preprocessing the PDF document containing this article using our previous `DEMOCRITUS` system. Example local causal models automatically constructed from this article are shown in the figure.

`DEMOCRITUS` constructs a detailed written summary of the causal claims both within the document, and the surrounding context that is automatically produced during the run. A brief excerpt of such a report is given below.

This note is a human-readable interpretation of the
Democritus credibility analysis.

What appears most credible

- upright walking in human ancestors --causes--> structural reorganization of the pelvis to support bipedal locomotion
- upright walking in human ancestors --increases--> the width and

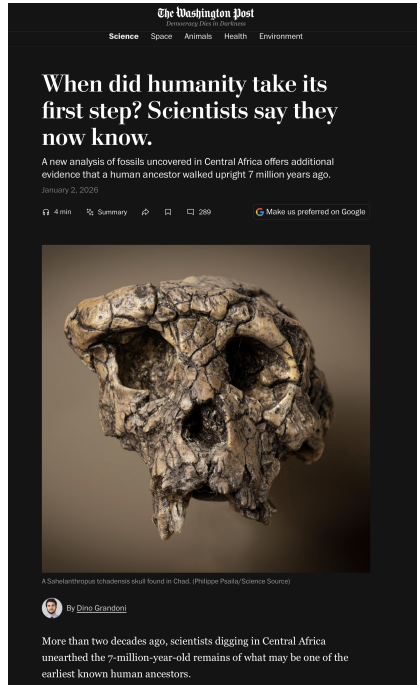


Figure 51: An article in The Washington Post on the origins of human bipedal walking.

- shortening of the pelvis to support bipedal locomotion
- upright walking in human ancestors --causes--> the development of an s-shaped spine to balance the body over the pelvis

In this chapter, we instead convert the local causal models, such as shown in the figure into a causal database over which SQL-like queries can be performed. We detail the CSQL data model next.

CSQL Data Model

Overview

CSQL represents causal knowledge extracted from document collections as a relational database. Rather than storing isolated facts or entity relations, the database encodes *causal structure* together with empirical support aggregated across many local causal models. The data model is designed to satisfy three requirements: (i) it must be queryable using standard SQL, (ii) it must preserve uncertainty and disagreement present in the source documents, and (iii) it must support compositional causal queries such as influence ranking, hub detection, and multi-step causal paths.

Core Relations

A CSQL database consists of the following core relations.

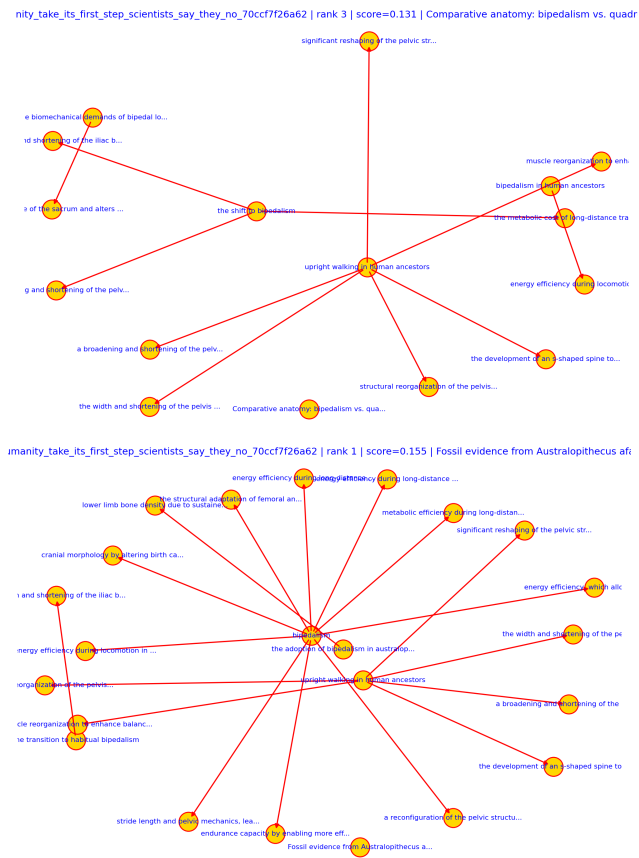


Figure 52: Two causal models constructed from The Washington Post article on the origins of bipedal walking by human ancestors 7 million years ago. DEMOCRITUS constructs thousands of such models automatically from the PDF document, which are then converted by CSQL into a causal database.

NODES. Each distinct causal concept is represented as a node:

node_id	64-bit identifier
label_canon	Canonicalized concept label
label_examples	Example surface forms
deg_in	Number of incoming causal edges
deg_out	Number of outgoing causal edges

EDGES. Causal relations are represented as weighted directed edges:

edge_id	Unique edge identifier
src_id	Source node
dst_id	Destination node
rel_type	Relation type (e.g., CAUSES, INFLUENCES)
polarity	Increase / decrease / unknown
support_lcms	Number of supporting local models
support_docs	Number of supporting documents
score_sum	Aggregated credibility mass
score_mean	Mean support per model
score_max	Maximum single-model score

EDGE SUPPORT.

Fine-grained provenance is captured in a support table:

edge_id	Referenced causal edge
doc_id	Source document
lcm_instance_id	Supporting local model
score_raw	Raw model score
coupling	Model-specific coupling strength

Derived Relations

CSQL additionally materializes derived relations computed from the core tables. These include:

CAUSAL MODULES (SCCs). Strongly connected components (SCCs) are computed over the causal graph:

scc_id	Component identifier
n_nodes	Number of nodes
n_edges	Number of edges
support_docs	Number of supporting documents
top_nodes	High-degree nodes in the component

Query Semantics

CSQL enables causal analysis through standard SQL queries. For example:

- **Backbone extraction:** Identify the most credible causal relations by ordering edges by `score_sum`.
- **Causal hubs:** Identify nodes with large outgoing score mass.
- **Downstream influence:** Join edges transitively to analyze multi-step causal paths.
- **Disagreement analysis:** Compare `score_max` and `score_mean` to detect fragile claims.

All such queries are executed using deterministic SQL over materialized tables, rather than neural inference or prompt-based generation.

Querying Causal Databases with CSQL

Once documents have been compiled into a CSQL database, causal analysis is performed using standard SQL queries. Unlike neural or prompt-based systems, CSQL queries execute deterministically over materialized relations that encode causal structure, uncertainty, and provenance.

This section illustrates several representative query patterns supported by CSQL. All examples are executed using unmodified SQL over Parquet-backed tables and can be run in any modern analytical database engine (e.g., DuckDB).

Backbone Extraction

A common analytic task is to identify the most credible causal relationships across the document collection. In CSQL, this corresponds to ordering causal edges by aggregated support.

```
SELECT
  e.rel_type,
  n1.label_canon AS src,
  n2.label_canon AS dst,
  e.support_lcms,
  e.score_sum
FROM atlas_edges e
JOIN atlas_nodes n1 ON e.src_id = n1.node_id
JOIN atlas_nodes n2 ON e.dst_id = n2.node_id
ORDER BY e.score_sum DESC
LIMIT 20;
```

This query returns a *causal backbone*: relationships that recur across many high-scoring local causal models. In the human bipedalism domain, this query surfaces relationships such as *bipedalism influencing metabolic efficiency* and *bipedalism increasing locomotion efficiency*, which are consistently supported across the extracted discourse.

Causal Hubs

CSQL supports identifying causal hubs: variables whose downstream influence is large when aggregated across models.

```
SELECT
  n.label_canon AS src,
  SUM(e.score_sum) AS out_mass,
  COUNT(*) AS out_degree
FROM atlas_edges e
JOIN atlas_nodes n ON e.src_id = n.node_id
GROUP BY n.node_id, n.label_canon
ORDER BY out_mass DESC
LIMIT 10;
```

This query ranks variables by the total credibility mass of their outgoing edges. In practice, this identifies dominant explanatory drivers within a domain. For example, in the bipedal walking corpus, *bipedalism* emerges as a central hub with broad downstream effects on energetics, skeletal structure, and locomotion.

Local Mechanism Exploration

CSQL enables focused exploration of a single causal variable by filtering on a specific source concept.

```
SELECT
  e.rel_type,
  n2.label_canon AS effect,
  e.support_lcms,
  e.score_sum
FROM atlas_edges e
JOIN atlas_nodes n1 ON e.src_id = n1.node_id
JOIN atlas_nodes n2 ON e.dst_id = n2.node_id
WHERE n1.label_canon = 'bipedalism'
ORDER BY e.score_sum DESC;
```

This query enumerates the strongest downstream effects of a given cause, ordered by credibility. Such queries are useful for drilling into mechanistic hypotheses associated with a particular concept, and can be composed with additional filters (e.g., relation type or polarity).

Provenance and Auditability

Every causal edge in CSQL retains provenance information linking it to supporting documents and local causal models.

```

SELECT
  s.doc_id,
  s.lcm_instance_id,
  s.score_raw,
  s.coupling
FROM atlas_edge_support s
WHERE s.edge_id = 10262796833405321330;

```

This query reveals exactly which documents and local models contributed to a given causal claim. Such provenance queries enable auditing, filtering by source, and sensitivity analysis under alternative aggregation strategies.

Cycles and Feedback Structures

CSQL supports detection of feedback and cyclic influence via derived relations such as strongly connected components (SCCs).

```

SELECT
  scc_id,
  n_nodes,
  n_edges,
  support_docs,
  top_nodes
FROM atlas_scc
ORDER BY n_nodes DESC
LIMIT 5;

```

Strongly connected components identify tightly coupled causal modules that may represent feedback loops, co-evolving variables, or higher-level mechanisms. These structures are difficult to express in DAG-based systems but arise naturally in language-derived causal discourse.

Example of a CSQL Database

The figure illustrates how CSQL represents causal knowledge extracted from a document collection as a relational database, using the example of upright walking in human ancestors. Each row in the `atlas_edges` table corresponds to a causal relation discovered across multiple local causal models (LCMs), aggregated over the discourse frame induced by the source articles.

In this example, *bipedalism* emerges as a high-degree source node. The table lists extracted claims about metabolic efficiency, energy efficiency, and endurance. An edge can aggregate repeated support across generated local causal models (LCMs), reflected in `support_lcms` and `score_sum`. These LCMs are not independent studies: in the checked ATLAS examples the leading rows generally have `support_docs=1`. Higher LCM support

therefore measures redundancy within the compilation process, not independent empirical replication.

Crucially, CSQL exposes this causal structure through standard SQL. For example, a query selecting edges with maximal `score_sum` directly answers questions such as: What are the strongest causal influences on bipedalism? Similarly, grouping by source nodes and summing outgoing score mass identifies causal hubs with large downstream impact. These operations require no specialized causal language or graph query engine—only relational joins and aggregations.

This example highlights the central design goal of CSQL: to make text-extracted causal claims accessible through familiar database abstractions. Rather than returning only a flat list of triples, CSQL materializes a fixed relational schema in which provenance, redundancy, polarity, and graph roles can be queried and composed using SQL. These are properties of the claim graph; they are not certificates of causal identification.

These examples illustrate how CSQL transforms document collections into relational databases of causal claims. By exposing causal structure, uncertainty, and provenance through standard SQL, CSQL enables analysts to perform causal exploration, auditing, and hypothesis generation using familiar database tools.

Quantitative Properties of CSQL Databases

Beyond supporting causal SQL queries, CSQL induces a structured relational database whose graph-theoretic and statistical properties can be summarized compactly. These properties are useful for (i) diagnosing document-induced topic drift, (ii) comparing domains, and (iii) choosing thresholds for downstream applications (e.g., filtering low-support edges or selecting “hub” concepts for exploration).

SCHEMA-LEVEL SUMMARY. For each document collection, CSQL produces two primary relations: (i) `atlas_nodes` (canonicalized concept objects) and (ii) `atlas_edges` (canonicalized causal generators). Edges carry both *support* (how many local causal models contain the edge) and *mass* (e.g., `score_sum`, `score_mean`, `score_max`) aggregated over model instances. This provides a principled basis for ranking edges without imposing a fixed ontology.

HEAVY-TAILED SUPPORT AND SCORE MASS. Across domains we observe a consistently heavy-tailed distribution over edge support and score mass: a small number of edges account for a disproportionate fraction of total score mass, while the majority of edges have low support and low mass. This is expected in discourse-derived causal graphs: a few mechanisms are stated (or paraphrased) redundantly across discourse neighborhoods, while

many peripheral claims occur only rarely. In practice, this means aggressive compression is possible: for many analytic tasks, retaining only the top k edges by `score_sum` preserves most of the interpretable backbone.

HUB CONCENTRATION. CSQL makes it explicit that influence is concentrated around a small set of hubs. Define the *outgoing mass* of a node x as

$$M_{\text{out}}(x) = \sum_{(x \xrightarrow{r} y) \in E} \text{score\_sum}(x, r, y).$$

In our bipedalism case study, the top hub (`bipedalism`) dominates outgoing mass by a wide margin. We quantify this concentration using simple concentration ratios (e.g., top-1 and top-5 share of total outgoing mass).

RELATION-TYPE MIX AND POLARITY MASS. Edges are normalized into a small set of relation types (e.g., `CAUSES`, `INFLUENCES`, `INCREASES`, `REDUCES`) and polarity buckets (`inc`, `dec`, `unk`). We report the distribution of relation types and polarity mass as a coarse indicator of discourse framing. For example, some domains emphasize `INFLUENCES/AFFECTS` (soft, explanatory discourse), while others emphasize `CAUSES` or signed relations (`INCREASES/REDUCES`).

CYCLES AND STRONGLY CONNECTED COMPONENTS. Unlike DAG-only representations, the induced causal database is not constrained to be acyclic. We compute strongly connected components (SCCs) over the atlas edges to identify feedback loops and closed influence motifs. At the document scale, SCCs may be sparse; at corpus scale, SCC modules become a useful signal for non-trivial causal connectivity and potential intervention loops. We store SCC summaries in `atlas_scc` when available.

Claim-graph ablation via SQL view rewriting

CSQL supports deterministic rewrites of the atlas edge relation. In the simplest *source ablation*, all outgoing claim edges with source X are removed and the query result is compared with the baseline view. This answers a database question—which reported paths and rankings depend on treating X as a source?—but it is not $\text{do}(X = x)$. In an SCM, a hard intervention replaces the mechanism for X , graphically cuts incoming arrows to X , and propagates through specified structural equations. The claim database supplies neither those equations nor an identified interventional distribution.

rel	dst (baseline top)	support	score_sum
INFLUENCES	metabolic efficiency during long-distance travel	15	1.462
INCREASES	energy efficiency during locomotion in early hominins	13	1.328
INCREASES	energy efficiency during long-distance locomotion	13	1.264
INFLUENCES	endurance capacity (heat dissipation, etc.)	13	1.256
INCREASES	energy efficiency (metabolic resources)	12	1.209
INFLUENCES	structural adaptation of femoral/tibial bones	11	1.201
INFLUENCES	stride length and pelvic mechanics	11	1.103

Source ablation: these edges vanish from the atlas view by construction.

A weighted source-sensitivity view can likewise downweight selected outgoing claim scores. This is sometimes suggestive of a soft intervention, but remains a database perturbation unless linked to a causal mechanism model:

```
WITH soft_do AS (
  SELECT
    e.*,
    CASE WHEN n1.label_canon='bipedalism'
      THEN 0.2*e.score_sum
    ELSE e.score_sum END AS score_sum_do
  FROM read_parquet('atlas_edges.parquet') e
  JOIN read_parquet('atlas_nodes.parquet') n1
    ON e.src_id=n1.node_id
)
SELECT ... ORDER BY score_sum_do DESC LIMIT 10;
```

Quantitative Summary of a CSQL Atlas

To complement the qualitative examples, we report summary statistics of the CSQL atlas constructed from the Washington Post document on the origins of bipedal walking. These statistics are computed directly over the atlas Parquet tables using DuckDB SQL queries. The statistics are reported in the table.

ATLAS SIZE. The atlas contains 501 canonical concept nodes and 65 unique canonical causal edges. In addition, the edge-support table contains 717 rows, where each row records an instance of an atlas edge being supported by a particular local causal model (LCM) in a particular document. This separation between (i) *unique canonical edges* and (ii) *support instances* is essential: the atlas table stores the distilled causal backbone, while the edge-support table retains provenance for auditability and reproducibility.

HUB CONCENTRATION. We measure hubness by aggregating outgoing causal mass per source concept (`score_sum` aggregated over outgoing edges). In this domain, the top hub is *bipedalism*. Remarkably, the top hub

Table 35: **Database diff for a source ablation.** Baseline: strongest outgoing generators from bipedalism. After source ablation the outgoing claim edges are removed by construction, so these rows disappear from query results; no counterfactual effect is estimated.

CSQL Atlas Summary: WaPo Human Origins (Bipedalism)		
Atlas Size		
Number of nodes (concepts)		501
Number of unique causal edges		65
Edge-support rows (LCM evidence)		717
Hub Concentration		
Top hub		<i>bipedalism</i>
Top hub outgoing mass share		0.748
Top-5 hubs outgoing mass share		0.857
Edge Weight Distribution (score_sum)		
Median (p_{50})		0.0337
90th percentile (p_{90})		0.847
99th percentile (p_{99})		1.377
Tail ratio (p_{99} / p_{50})		40.8
Relation-Type Mass Breakdown		
Relation type	Polarity	Total score mass
INFLUENCES	unk	6.19
INCREASES	inc	5.13
CAUSES	unk	0.30
REDUCES	dec	0.14
LEADS_TO	unk	0.03
AFFECTS	unk	0.01

alone accounts for approximately 74.8% of the total outgoing score mass in the atlas, and the top five hubs account for 85.7%. This demonstrates a strongly centralized causal atlas in this document-induced discourse frame: most highly supported causal relations radiate from a small number of discourse pivots.

HEAVY-TAILED EDGE STRENGTH. Edge strengths in CSQL are strongly heavy-tailed. Using `score_sum` as the edge weight, the median (50th percentile) edge mass is 0.0337, while the 90th percentile is 0.847, and the 99th percentile is 1.377. The ratio $p_{99} / p_{50} \approx 40.8$ quantifies the extreme concentration: a small number of edges dominate the causal backbone, while most edges carry comparatively small mass. This is consistent with the empirical behavior observed in Democritus v2.0: causal discourse induces a large hypothesis space, but only a small fraction of hypotheses are redundantly supported.

RELATION-TYPE AND POLARITY MASS. We also summarize atlas edges by normalized relation type (`rel_type`) and polarity (`polarity`). In this atlas, most edges fall into two broad families: `INFLUENCES` (unknown polarity) and `INCREASES` (positive polarity). By mass, `INFLUENCES` con-

Table 36: Quantitative summary of the CSQL atlas induced from a Washington Post article on the origins of human bipedal walking. The atlas exhibits strong hub concentration, heavy-tailed edge weights, and a dominance of influence- and increase-type causal relations. All statistics are computed directly via SQL queries over Parquet-backed CSQL tables using DuckDB.

Quantity	Value
Number of nodes (concepts)	501
Number of causal edges	65
Number of edge-support records	717

tributes 6.19 units of score mass, while *INCREASES* contributes 5.13, with smaller contributions from *CAUSES*, *REDUCES*, and a handful of miscellaneous relations. This breakdown reflects the linguistic character of the source document and the implicit uncertainty of causal discourse: journalistic and paleoanthropological claims are often phrased as influences and associations rather than signed, quantified effects.

TOP EDGES. Finally, we list the top weighted edges (by `score_sum`). The highest-mass edges take *bipedalism* as the source and connect it to mechanistic consequences such as metabolic efficiency, endurance capacity, stride length, and locomotion energy efficiency. These edges collectively form the backbone of the induced atlas.

TAKEAWAY. These quantitative summaries illustrate why a CSQL atlas is useful: the induced schema is not merely a flat list of extracted triples, but a weighted causal database with (i) strong hub structure, (ii) heavy-tailed edge strength, and (iii) an explicit decomposition by relation-type and polarity. Such structure is immediately queryable by standard SQL, enabling downstream analytics (e.g., identifying dominant drivers, tracing multi-step influence chains, or locating high-uncertainty edges) without requiring a hand-designed ontology or fixed domain schema.

In summary, CSQL yields not only a queryable causal database but also a set of stable quantitative signals (heavy tails, hub concentration, relation-type mixtures) that support corpus diagnostics and downstream system design.

Database Scale and Structure

The table summarizes the size of the causal database constructed from a single Washington Post article on the origins of human bipedal walking. Despite originating from a single document, the induced discourse frame expands into hundreds of causal variables and hundreds of supported causal relations.

Table 37: Scale of the CSQL causal database constructed from a single document. Edge-support rows correspond to individual local causal models (LCMs) that support each aggregated causal edge.

(a) Backbone edges (top by score mass)

```
SELECT e.rel_type, n1.label_canon AS src, n2.label_canon AS dst,
       e.support_lcms, e.score_sum
FROM read_parquet('atlas_edges.parquet') e
JOIN read_parquet('atlas_nodes.parquet') n1 ON e.src_id=n1.node_id
JOIN read_parquet('atlas_nodes.parquet') n2 ON e.dst_id=n2.node_id
ORDER BY e.score_sum DESC
LIMIT 8;
```

```
INFLUENCES | bipedalism | metabolic efficiency ... | 15 | 1.4624
INCREASES | bipedalism | energy efficiency ... | 13 | 1.3282
INCREASES | bipedalism | energy efficiency ... | 13 | 1.2644
INFLUENCES | bipedalism | endurance capacity ... | 13 | 1.2557
INCREASES | bipedalism | energy efficiency ... | 12 | 1.2093
INFLUENCES | bipedalism | structural adaptation... | 11 | 1.2010
INFLUENCES | bipedalism | stride length ... | 11 | 1.1029
INFLUENCES | reduced forest cover | selection ... | 6 | 0.4637
```

(b) Causal hubs (sources with largest downstream mass)

```
SELECT n.label_canon AS src, SUM(e.score_sum) AS out_mass, COUNT(*) AS out_deg
FROM read_parquet('atlas_edges.parquet') e
JOIN read_parquet('atlas_nodes.parquet') n ON e.src_id=n.node_id
GROUP BY n.node_id, n.label_canon
ORDER BY out_mass DESC
LIMIT 8;
```

```
bipedalism | 8.8239 | 10
reduced forest cover | 0.4637 | 1
changes in environmental conditions ... | 0.4026 | 1
abnormal femoral tubercle position | 0.2853 | 2
fossil structure of ardipithecus | 0.1358 | 1
muscle reorganization for balance ... | 0.1283 | 1
larger brain size in primates | 0.0814 | 1
lateral displacement of the femoral ... | 0.0763 | 1
```

(c) Local mechanism: top outgoing edges from a hub

```
SELECT e.rel_type, n2.label_canon AS dst, e.support_lcms, e.score_sum
FROM read_parquet('atlas_edges.parquet') e
JOIN read_parquet('atlas_nodes.parquet') n1 ON e.src_id=n1.node_id
JOIN read_parquet('atlas_nodes.parquet') n2 ON e.dst_id=n2.node_id
WHERE n1.label_canon='bipedalism'
ORDER BY e.score_sum DESC
LIMIT 7;
```

```
INFLUENCES | metabolic efficiency ... | 15 | 1.4624
INCREASES | energy efficiency ... | 13 | 1.3282
INCREASES | energy efficiency ... | 13 | 1.2644
INFLUENCES | endurance capacity ... | 13 | 1.2557
INCREASES | energy efficiency ... | 12 | 1.2093
INFLUENCES | structural adaptation... | 11 | 1.2010
INFLUENCES | stride length ... | 11 | 1.1029
```

Figure 53: CSQL in action (DuckDB). Example SQL queries over a Parquet-backed causal atlas. All results are deterministic given the compiled atlas tables.

```

-- Baseline: top outgoing generators from the hub "bipedalism"
SELECT
    e.rel_type ,
    n2.label_canon AS dst ,
    e.support_lcms ,
    e.score_sum
FROM read_parquet('atlas_edges.parquet') e
JOIN read_parquet('atlas_nodes.parquet') n1 ON e.src_id = n1.node_id
JOIN read_parquet('atlas_nodes.parquet') n2 ON e.dst_id = n2.node_id
WHERE n1.label_canon = 'bipedalism'
ORDER BY e.score_sum DESC
LIMIT 10;

-- Source ablation: remove outgoing claim edges from "bipedalism"
WITH intervened_edges AS (
    SELECT e.*
    FROM read_parquet('atlas_edges.parquet') e
    JOIN read_parquet('atlas_nodes.parquet') n1 ON e.src_id = n1.node_id
    WHERE n1.label_canon <> 'bipedalism'
)
SELECT
    e.rel_type ,
    n1.label_canon AS src ,
    n2.label_canon AS dst ,
    e.support_lcms ,
    e.score_sum
FROM intervened_edges e
JOIN read_parquet('atlas_nodes.parquet') n1 ON e.src_id = n1.node_id
JOIN read_parquet('atlas_nodes.parquet') n2 ON e.dst_id = n2.node_id
ORDER BY e.score_sum DESC
LIMIT 10;

```

Figure 54: **CSQL claim-graph ablation as SQL view rewriting.** Left: baseline query retrieves the strongest outgoing causal generators from the hub *bipedalism*. Right: a source-ablation view removes all edges whose source is *bipedalism*, revealing the next-ranked hubs and claims in the atlas. This is a sensitivity query, not an identified causal intervention.

Hub Dominance and Causal Centralization

Causal influence in the atlas is highly concentrated around a small number of hubs. We quantify this concentration by measuring the fraction of total outgoing causal mass attributable to the most influential source nodes.

Top hub	Top-1 mass share	Top-5 mass share
bipedalism	0.748	0.857

Table 38: Causal hub dominance in the CSQL atlas. The single most influential node (bipedalism) accounts for nearly 75% of outgoing causal mass, while the top five hubs account for over 85%.

Heavy-Tailed Causal Strength

Causal edge strength in CSQL follows a strongly heavy-tailed distribution. We characterize this using empirical quantiles of the aggregated edge score distribution.

p_{50}	p_{90}	p_{99}	p_{99}/p_{50}
0.034	0.847	1.377	40.8

Table 39: Quantiles of causal edge score distribution. The extreme p_{99}/p_{50} ratio confirms a strongly heavy-tailed energy landscape over causal hypotheses.

Relation-Type Composition

The causal database contains a heterogeneous mix of relation types, with asymmetric distribution of causal mass across predicates.

Relation type	# edges	Total mass
INFLUENCES	19	6.19
INCREASES	27	5.13
CAUSES	9	0.30
REDUCES	7	0.14
LEADS_TO	1	0.03
AFFECTS	2	0.01

The CSQL Data Model

CSQL represents causal claims extracted from document collections as a small set of relational tables with well-defined semantics. The core table schema is fixed by design; language processed by the DEMOCRITUS pipeline induces the node vocabulary, edge instances, scores, and provenance records that populate it.

At a high level, CSQL consists of four core relations:

- `atlas_nodes`: canonical causal concepts,
- `atlas_edges`: aggregated causal relations,
- `atlas_edge_support`: provenance linking edges to local causal models and documents,
- `atlas_scc`: strongly connected components capturing feedback structure.

Together, these tables define a causally meaningful relational schema that supports both structural analysis and evidence-aware querying using standard SQL.

Nodes: Canonical Causal Concepts

Each row in `atlas_nodes` represents a canonicalized causal concept induced from language, such as *bipedalism*, *energy efficiency during locomotion*, or *metabolic efficiency during long-distance travel*. Nodes are obtained by normalizing and de-duplicating subject and object phrases across all extracted discourse triples.

Key columns include:

- `node_id`: a stable identifier,
- `label_canon`: canonical concept string,
- `label_examples`: representative surface forms,
- `deg_in`, `deg_out`: in- and out-degrees in the causal atlas.

Nodes function as the *objects* of the induced causal structure. No external ontology or controlled vocabulary is required; the object space is learned directly from text.

Table 40: Distribution of causal relation types. Influence-style relations dominate both numerically and in total causal mass, while strict causal predicates are rarer and more conservative.

Edges: Aggregated Causal Relations

Each row in `atlas_edges` represents a causal relation aggregated across many local causal models (LCMs). An edge corresponds to a canonical triple

$$(\text{src\_id}, \text{rel\_type}, \text{dst\_id}),$$

where `rel_type` ranges over normalized relations such as `INCREASES`, `REDUCES`, `CAUSES`, or `INFLUENCES`.

In addition to structural information, each edge carries quantitative evidence:

- `support_lcms`: number of LCMs in which the edge appears,
- `support_docs`: number of documents contributing evidence,
- `score_sum`, `score_mean`, `score_max`: aggregated credibility scores,
- polarity mass fields distinguishing increasing, decreasing, and unknown effects.

These values encode *agreement under variation*: edges that recur across many high-scoring hypotheses accumulate greater score mass and therefore higher causal credibility.

Edge Support and Provenance

CSQL preserves fine-grained provenance via the `atlas_edge_support` table, which links each aggregated edge back to:

- specific LCM instances,
- source documents,
- raw per-model scores and coupling values.

This design enables auditability and traceability. Users can drill down from a high-level causal claim to the specific models and textual evidence that support it.

Strongly Connected Components

The `atlas_scc` table captures strongly connected components of the causal atlas graph. These components represent feedback loops or tightly coupled causal subsystems, which cannot be represented in DAG-based causal formalisms.

SCCs provide a natural entry point for reasoning about cyclic causality, mutual reinforcement, and dynamical regimes—key motivations for moving beyond DAGs to more expressive causal representations.

Causal Reasoning as SQL

A central contribution of CSQL is that *causal reasoning can be expressed directly as SQL queries* over the induced schema. No specialized graph query language or causal DSL is required.

Identifying Causal Backbones

The strongest causal claims in a document collection correspond to edges with maximal aggregated score mass. For example:

```
SELECT
  e.rel_type,
  n1.label_canon AS src,
  n2.label_canon AS dst,
  e.support_lcms,
  e.score_sum
FROM atlas_edges e
JOIN atlas_nodes n1 ON e.src_id = n1.node_id
JOIN atlas_nodes n2 ON e.dst_id = n2.node_id
ORDER BY e.score_sum DESC
LIMIT 50;
```

This query extracts the *causal backbone* of the discourse frame—relations that survive across many high-scoring hypotheses.

Causal Hubs and Downstream Influence

Causal hubs are nodes with large outgoing score mass:

```
SELECT
  n.label_canon AS src,
  SUM(e.score_sum) AS out_mass
FROM atlas_edges e
JOIN atlas_nodes n ON e.src_id = n.node_id
GROUP BY n.label_canon
ORDER BY out_mass DESC;
```

In the upright walking example, *bipedalism* emerges as the dominant hub, with strong downstream influence on metabolic efficiency, endurance, and locomotion.

Causal Composition

Multi-step causal chains correspond to relational joins:

```
SELECT
```

```

    n1.label_canon AS a,
    e1.rel_type AS r1,
    n2.label_canon AS b,
    e2.rel_type AS r2,
    n3.label_canon AS c,
    (e1.score_sum + e2.score_sum) AS path_score
FROM atlas_edges e1
JOIN atlas_edges e2 ON e1.dst_id = e2.src_id
JOIN atlas_nodes n1 ON e1.src_id = n1.node_id
JOIN atlas_nodes n2 ON e1.dst_id = n2.node_id
JOIN atlas_nodes n3 ON e2.dst_id = n3.node_id
ORDER BY path_score DESC;

```

Such queries implement causal composition directly in SQL.

Cycles and Feedback

Cycles are detected via self-joins or SCC tables. For example, mutual influence:

```

SELECT
    n1.label_canon AS a,
    e1.rel_type AS r1,
    n2.label_canon AS b,
    e2.rel_type AS r2
FROM atlas_edges e1
JOIN atlas_edges e2
    ON e1.src_id = e2.dst_id
    AND e1.dst_id = e2.src_id
JOIN atlas_nodes n1 ON e1.src_id = n1.node_id
JOIN atlas_nodes n2 ON e1.dst_id = n2.node_id;

```

This enables explicit querying of feedback structures absent from DAG-based causal systems.

COUNTERFACTUAL QUERY VIA SQL VIEW REWRITING. CSQL supports source-ablation queries by deterministically removing selected rows from the atlas edge relation. The difference between baseline and ablated views reveals which reported paths and rankings depend on the removed source. This is graph sensitivity analysis. It should not be called a hard intervention or a counterfactual unless an external causal model supplies mechanism-replacement and propagation semantics.

Quantitative Summary of the CSQL Corpus Database

The table reports corpus-level summary statistics for the merged CSQL database constructed from multiple document-derived atlases. The database

contains canonicalized concept nodes and relation edges, together with provenance (edge-support rows linking edges to document and local-model instances) and aggregate credibility statistics (e.g., score mass and polarity mass). The resulting score distribution is heavy-tailed, and the outgoing score mass is concentrated in a small number of causal hubs, reflecting the system’s tendency to surface central drivers in the discourse manifold.

Quantity	Value
Canonical nodes ($ \text{nodes} $)	839
Canonical edges ($ \text{edges} $)	450
Edge-support rows ($ \text{edge_support} $)	1311
Documents ($ \text{docs} $)	10
Atlases merged ($ \text{atlases} $)	7

Table 41: Corpus-level size and provenance counts for the merged CSQL database.

Hub source (src)	Out mass	Out edges	LCM support
exposure to roundup	105.03	3	24
dark chocolate consumption	64.70	5	20
severity of post impacts	44.76	2	17
impact of researchers	33.68	3	15
application of roundup	30.58	2	9

Table 42: Highest-scoring source hubs in the merged CSQL database.

rel_type	polarity	#edges	score mass
INFLUENCES	unk	135	353.78
LEADS_TO	unk	80	188.11
REDUCES	dec	75	181.83
INCREASES	inc	89	156.30
CAUSES	unk	50	89.99
AFFECTS	unk	21	75.71

Table 43: Distribution of edge mass by normalized relation type and polarity in the merged corpus.

INTERPRETATION. The merged CSQL database exhibits two recurring properties. First, the aggregate credibility signal (score mass) is heavy-tailed: a small fraction of edges and sources account for a large fraction of total mass. Second, this mass is organized around a handful of causal hubs, which act as central drivers in the discourse manifold (e.g., *exposure to roundup* and *dark chocolate consumption*). These hubs are precisely the nodes that become most useful for interactive causal querying: they support SQL queries that surface high-impact downstream consequences, compare competing mechanisms, and identify which claims are stable across multiple local causal models. Finally, the relation-mass breakdown shows that the induced schema concentrates into a small, reusable vocabulary of causal predicates (INFLUENCES/LEADS_TO/INCREASES/REDUCES), which is critical for portability across corpora without requiring a hand-built ontology.

Algorithmic Construction of CSQL Databases

This section describes the deterministic compilation step that turns a collection of local causal models (LCMs) into a *causal SQL database* (CSQL).

The input to this stage is a directory of LCMs produced by an upstream system (e.g., DEMOCRITUS) and optionally accompanied by scoring metadata (e.g., `scores.csv`). The output is a small set of Parquet tables that can be queried directly in DuckDB (or any Parquet-capable SQL engine).

Input Contract

We assume each document d has an associated set of LCMs stored as JSON files. An LCM is represented as a directed labeled multigraph $M = (V_M, E_M)$ with a focus string and optional metadata. Each edge is a record

$$e = (\text{src}, \text{rel}, \text{dst})$$

where `src` and `dst` are surface strings and `rel` is a free-form relation phrase (e.g., “increases”, “reduces”, “influences”). Optionally, each LCM is accompanied by a scalar score and auxiliary fields (radius, model size, coupling, etc.) used for aggregation.

CSQL Schema

CSQL materializes three core tables (plus one optional analytic table):

- **nodes**: canonicalized concepts (entities/variables) discovered from LCM node strings.
- **edges**: canonicalized causal edges aggregated across all LCMs (deduplicated).
- **edge_support**: provenance table linking each canonical edge to the document and LCM instances that support it (including local model scores).
- **scc** (optional): strongly connected component summaries over the induced graph, to expose cycles/modules.

This separation is deliberate and mirrors standard database normalization practice: `edges` provides a stable fact table for queries, while `edge_support` preserves provenance and enables auditing, filtering, and corpus merging.

Canonicalization and Keying

To make the database queryable and mergeable across documents, CSQL normalizes strings and relations. We define:

CANONICAL NODE LABELS. A normalization function $\text{canon}(\cdot)$ maps raw node strings to a canonical form (lowercasing, whitespace/punctuation normalization, dash unification, etc.). Canonical nodes are keyed by:

$$\text{node\_id} = H(\text{canon}(x))$$

where H is a stable 64-bit hash.

CANONICAL RELATION TYPES. A relation normalization function $\text{reltype}(\cdot)$ maps raw relation phrases to a small controlled vocabulary (e.g., CAUSES, INFLUENCES, INCREASES, REDUCES, AFFECTS, LEADS_TO). We also extract a coarse *polarity* label $\in \{\text{inc}, \text{dec}, \text{unk}\}$.

CANONICAL EDGES. A canonical edge is identified by:

$$\text{edge\_id} = H(\text{src\_id} \parallel \text{rel\_type} \parallel \text{dst\_id})$$

This provides stable deduplication within and across documents.

Atlas Builder: From LCMs to Parquet

Algorithm 15 describes the CSQL compilation procedure. The output tables store: (i) canonical objects and arrows (`nodes`, `edges`), (ii) provenance and scoring (`edge_support`), and (iii) optional graph analytics (`scc`).

Algorithm 15: BUILDATLAS: Compile LCMs into a CSQL database

Input: Document runs root directory $\mathcal{R}$ containing per-document LCM folders; optional scoring metadata (`scores.csv`); min-edge filter τ ; relation whitelist/blacklist.

Output: Parquet tables: `nodes.parquet`, `edges.parquet`, `edge_support.parquet` (and optional `scc.parquet`).

```

1: Initialize empty maps: NodeMap : node_id  $\mapsto$  node record, EdgeAgg :
   edge_id  $\mapsto$  aggregate record
2: Initialize empty list SupportRows
3: for all documents  $d$  under  $\mathcal{R}$  do
4:   Load set of LCM JSON files  $\{M_i\}$  for  $d$  (optionally filtered by score/size)
5:   for all LCMs  $M_i$  do
6:     Extract metadata: doc_id, lcm_instance_id, score,
       score_raw, coupling
7:     for all edges  $e = (u, r, v) \in E_{M_i}$  do
8:       if  $|E_{M_i}| < \tau$  then continue
9:       end if
10:       $u_c \leftarrow \text{canon}(u)$ ;  $v_c \leftarrow \text{canon}(v)$ 
11:       $\text{src\_id} \leftarrow H(u_c)$ ;  $\text{dst\_id} \leftarrow H(v_c)$ 
12:       $(\text{rel\_type}, \text{polarity}) \leftarrow \text{reltype}(r)$ 
13:       $\text{edge\_id} \leftarrow H(\text{src\_id} \parallel \text{rel\_type} \parallel \text{dst\_id})$ 
14:      Update NodeMap with examples for  $u, v$ 
15:      Update EdgeAgg[edge_id]: support counts, score mass sums,
       polarity mass
16:      Append support row to SupportRows:
17:
       (edge_id, doc_id, atlas_id, lcm_instance_id, score, score_raw, coupling)
18:     end for
19:   end for
20: end for
21: Materialize nodes table from NodeMap with degree stats computed from
   edges
22: Materialize edges table from EdgeAgg with summary fields:
       support_lcms, support_docs, score_sum, score_mean,
       score_max, pol_mass_inc, pol_mass_dec, pol_mass_unk, controversy
23: Materialize edge_support table from SupportRows
24: (Optional) Build induced directed graph from edges; compute SCCs; write
   scc.parquet
25: Write all tables to Parquet
26: return

```

AGGREGATION AND “CONTROVERSY”. Each canonical edge aggregates evidence across many LCM instances. Let w_i be the score mass contributed by LCM i to a canonical edge (we use the adjusted score when available). For

polarity mass we define:

$$m_{\text{inc}} = \sum_{i:\text{polarity}=\text{inc}} w_i, \quad m_{\text{dec}} = \sum_{i:\text{polarity}=\text{dec}} w_i, \quad m_{\text{unk}} = \sum_{i:\text{polarity}=\text{unk}} w_i.$$

A simple controversy indicator is then:

$$\text{controversy} = \frac{\min(m_{\text{inc}}, m_{\text{dec}})}{m_{\text{inc}} + m_{\text{dec}} + \epsilon}.$$

This yields $\text{controversy} = 0$ when only one direction is supported and approaches 0.5 when inc/dec are balanced.

Corpus Merge: Union of CSQL Databases

Because all identifiers are canonical (hash-based), merging multiple atlas databases is a deterministic union operation followed by re-aggregation. Algorithm 16 sketches the merge procedure.

Algorithm 16: MERGEATLASES: Merge per-document CSQL databases into a corpus database

Input: A directory containing atlas folders $\{\mathcal{A}_j\}$, each with nodes, edges, edge_support.

Output: Merged corpus CSQL database: nodes, edges, edge_support.

- 1: Load and concatenate all edge_support tables; optionally prefix doc_id with atlas id for disambiguation
 - 2: Group edge_support by edge_id to recompute edge aggregates (support counts, score sums, polarity mass)
 - 3: Construct merged edges from aggregated groups
 - 4: Construct merged nodes by collecting all referenced src_id, dst_id and recomputing degrees from merged edges
 - 5: Write merged tables to Parquet
 - 6: **return**
-

In this way, CSQL supports both *document-level* analysis and *corpus-level* analysis using the same schema. Crucially, provenance is preserved: corpus queries can be refined by filtering on doc_id, atlas_id, or lcm_instance_id, enabling traceable drill-down from aggregate causal claims back to local models and source documents.

Practical Notes for Users

CSQL is designed to be usable with off-the-shelf tools:

- Any SQL engine with Parquet support can query the tables; we use DuckDB.
- Most queries are expressed as joins between edges and nodes for human-readable labels.

- Provenance-aware queries join `edge_support` to recover document and LCM-level evidence.

This makes CSQL a *causal database backend* that can be placed under visualization layers, RAG systems, or higher-level reasoning systems (including DEMOCRITUS itself).

CSQL from RAG-Compiled Causal Corpora

A key design goal of CSQL is to decouple *causal discourse compilation* from *causal database construction*. While earlier sections focused on LLM-driven discourse compilation via DEMOCRITUS, the CSQL pipeline is agnostic to how causal claims are obtained. In this subsection, we show that CSQL can be constructed directly from *RAG-compiled causal corpora*, without requiring access to the original documents or an LLM-based generation pipeline.

We use the *Testing Causal Claims (TCC)* dataset³⁷ as a canonical example. TCC extracts causal claims from a large corpus of economics papers using information extraction and retrieval-based methods, yielding structured records of the form

(cause, effect, sign, method, document metadata).

Rather than treating TCC as a static knowledge graph, we interpret it as a *precompiled causal discourse layer*. CSQL then compiles this layer into a relational causal database supporting causal aggregation, ranking, and intervention-style queries.

DISCOURSE-TO-DATABASE COMPILATION. Given a RAG-compiled corpus such as TCC, CSQL performs a deterministic compilation step that maps extracted causal claims into a small set of normalized relational tables. Canonicalized cause and effect phrases become nodes, while typed causal relations become edges. Repeated claims across documents are aggregated, producing support counts and score mass analogous to those obtained from DEMOCRITUS-generated local causal models.

Importantly, CSQL does not assume a fixed ontology, predefined schema, or hand-designed knowledge graph. Instead, the schema is induced directly from the causal discourse itself. Provenance information—such as document identifiers, publication year, and causal inference method—is preserved in an auxiliary edge-support table, enabling downstream filtering and auditability.

UNIFIED CAUSAL QUERYING ACROSS COMPILATION METHODS. Once compiled, a CSQL database constructed from a RAG-based corpus supports the same class of causal queries as one constructed from DEMOCRITUS. For example:

³⁷ Prashant Garg and Thiemo Fetzer. Causal claims in economics, 2026. URL <https://arxiv.org/abs/2501.06873>

- identifying global causal hubs with the largest downstream influence,
- ranking causal claims by redundancy and aggregate support,
- detecting controversial claims with mixed directional evidence,
- approximating interventions by removing or conditioning on selected sources.

From the perspective of the query layer, there is no distinction between causal databases derived from LLM-generated discourse and those derived from RAG-compiled corpora.

IMPLICATIONS FOR SCALE AND INTEROPERABILITY. This separation enables CSQL to operate at corpus scale without requiring LLM inference over tens of thousands of documents. Large, curated resources such as TCC can be ingested directly, allowing CSQL to support causal analysis over orders of magnitude more documents than would be feasible with generative pipelines alone. At the same time, databases compiled from RAG-based sources and from DEMOCRITUS can be merged seamlessly, yielding a unified causal database spanning heterogeneous document collections.

Viewed abstractly, CSQL serves as a common compilation target for diverse causal discourse generators—LLMs, RAG systems, and classical information extraction pipelines—while providing a uniform, SQL-native interface for causal analysis.

CSQL from a RAG-Compiled Causal Corpus: Testing Causal Claims (TCC)

We also instantiate CSQL on the *Testing Causal Claims* (TCC) dataset, which already provides a corpus-level extraction of directed cause–effect claims from ~45K economics papers. The *Testing Causal Claims* (TCC) project³⁸ extracts and catalogs causal claims from large corpora of economics papers, producing a global index of cause–effect statements enriched with metadata such as sign, statistical significance, and inference method. In this setting, CSQL does not run the DEMOCRITUS discourse compiler; instead, it treats TCC as a pre-compiled discourse layer and compiles it into the same relational schema (nodes, edges, edge_support).

The resulting database contains 295,459 canonicalized concept strings and 260,777 distinct directed edges (265,656 edge-support rows), spanning 44 publication years. Because the current ingest uses the generic relation label INFLUENCES and does not yet project sign/method metadata into CSQL, edge mass largely tracks frequency (number of supporting papers), producing a near-degenerate heavy tail (e.g., median support mass 1, 99th percentile 2). This behavior is expected: TCC is already an edge list, whereas DEMOCRITUS-native atlases aggregate many overlapping local causal hypotheses and therefore induce richer relation vocabularies and heavier-tailed score mass.

³⁸ Prashant Garg and Thiemo Fetzer. Causal claims in economics, 2026. URL <https://arxiv.org/abs/2501.06873>

Despite this limitation, the CSQL interface already supports useful corpus-scale queries such as hub discovery and backbone extraction; in ongoing work we project TCC’s sign and identification-method metadata into CSQL to enable polarity-aware queries and method-conditioned disagreement analyses.

To stress-test CSQL at scale without running DEMOCRITUS over tens of thousands of PDFs, we treat the *Testing Causal Claims* (TCC) dataset as an already-compiled causal discourse layer. TCC provides paper-level causal edges (cause $\rightarrow$ effect) with metadata such as year and the stated causal inference method. We compile this corpus into a CSQL database by canonicalizing node strings (causes/effects) into concept identifiers, aggregating identical edges across papers, and storing per-paper evidence rows in an `edge_support` table. All statistics below were computed directly with DuckDB queries over the resulting Parquet tables.

SCALE. The compiled TCC–CSQL database contains **265,656** evidence rows (one per extracted claim instance), spanning **45,319** distinct papers, **44** publication years, and **1,575** distinct method strings (as reported in the corpus). At the schema level, the database contains **295,459** canonical concept nodes and **260,777** unique directed causal edges. This illustrates that CSQL can ingest a large RAG/IE-compiled causal corpus and expose it as a unified, queryable causal database without requiring a hand-designed ontology.

HUBS AND BACKBONE CLAIMS. A hallmark of corpus-scale causal discourse is the emergence of *hub variables* with large outgoing mass. In TCC–CSQL, the largest outgoing hubs include *education*, *trade liberalization*, *age*, *inflation*, and *monetary policy*, each participating in hundreds of distinct outgoing edges. For example, the query “What most strongly influences inflation?” returns recurrent claims such as *monetary policy* $\rightarrow$ *inflation*, *money growth* $\rightarrow$ *inflation*, and *output gap* $\rightarrow$ *inflation*, ranked by paper support. These results are not interpreted as ground-truth causality; rather, they operationalize the corpus as a causal claim space that can be searched, filtered, and audited.

TEMPORAL DYNAMICS. Because each claim instance in `edge_support` carries a publication year, CSQL supports time-sliced causal analysis. Aggregating claim instances by year reveals strong growth over time: the corpus contains under 2,000 extracted claim instances per year in the early 1980s, rising to over 10,000 per year in the late 2010s and above 18,000 in 2020. For a specific canonical claim (e.g., *monetary policy* $\rightarrow$ *inflation*), CSQL can return the distribution of supporting papers by year, enabling longitudinal tracking of how particular causal claims appear and reappear across decades.

METHOD STRINGS AND HETEROGENEITY. The TCC metadata includes

Quantity	Value
Claim instances (edge_support rows)	265,656
Unique papers (doc_id)	45,319
Unique years	44
Distinct method strings	1,575
Canonical concept nodes (nodes)	295,459
Unique causal edges (edges)	260,777

Hub (cause)	Outgoing mass	# outgoing edges
education	442	374
trade liberalization	391	370
age	358	339
inflation	280	250
monetary policy	270	211

a reported causal inference method field. In the current corpus snapshot, the method field contains a long tail of heterogeneous strings (e.g., IV, DID, TWFE, but also many fine-grained or free-text variants). This is reflected in the large number of distinct method values (1,575). CSQL treats the method column as an observational attribute: users can filter or group by methods as-is, or optionally normalize method strings into a smaller controlled taxonomy for downstream analysis.

Topos Theoretic CSQL: Red-Wine Pullbacks, Pushouts, and Ω

We now illustrate the application of Topos Causal Models (TCMs) introduced in the previous chapters to CSQL, by combining two studies of the health benefits of drinking red wine. The two document atlases are treated as CSQL objects over a shared claim-key base, and we materialize three universal constructions: an exact pullback, a soft pullback, and a pushout merge. The exact pullback is empty, showing that literal canonicalized claim keys do not coincide across the two document atlases. But a soft pullback over the shared *resveratrol*-as-source interface recovers two cross-atlas witness edges, which are then glued into the pushout.

TOPOS-THEORETIC READING. The key new point is that we can use subobject classifiers in TCMs to make the subobject-classifier pattern explicit at the database level. Let X_{rw} denote the pushout provenance object for the merged red-wine atlas. Then the study defines a finite truth object

$$\Omega_{rw} = \{\text{CONSENSUS}, \text{WEAK\_CONSENSUS}, \text{A\_ONLY}, \text{B\_ONLY}\},$$

together with a characteristic map

$$\chi_{rw} : X_{rw} \rightarrow \Omega_{rw}$$

that labels each merged edge by its cross-atlas status. The corresponding predicates, such as the consensus witness subobject $S_{\text{cons}} \hookrightarrow X_{rw}$, are

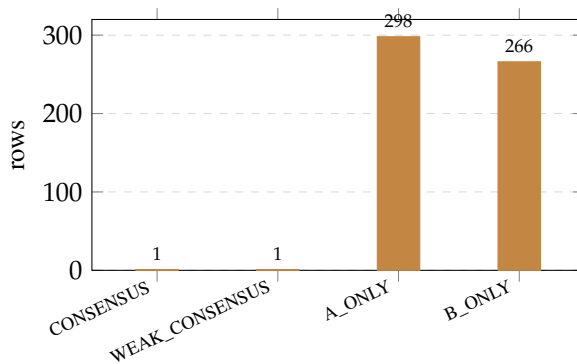
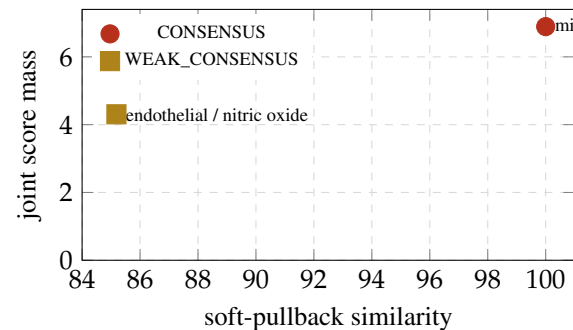
Table 44: Summary statistics for the TCC–CSQL database. The corpus is treated as a RAG/IE-compiled causal discourse layer; CSQL compiles it into a queryable causal database with edge aggregation and per-paper evidence.

Table 45: Top corpus hubs in TCC–CSQL, ranked by outgoing support mass (sum of support_docs over outgoing edges). These hubs summarize the dominant causal discourse centers in the economics corpus.

Claim affecting inflation	Support (papers)	Mass
monetary policy $\rightarrow$ inflation	15	15
money growth $\rightarrow$ inflation	12	12
output gap $\rightarrow$ inflation	11	11
monetary policy shocks $\rightarrow$ inflation	8	8
fiscal policy $\rightarrow$ inflation	7	7

Quantity	Value
Exact pullback rows	0
Soft pullback witness rows	2
Pushout rows	566
CONSENSUS rows	1
WEAK_CONSENSUS rows	1
A_ONLY rows	298
B_ONLY rows	266

Ω label	Witness mechanism	sim.	joint score
CONSENSUS	<i>resveratrol</i> $\rightarrow$ mitochondrial biogenesis, enhancing cardiac energy metabolism and reducing susceptibility to ischemic injury	100.0	6.892
WEAK_CONSENSUS	<i>resveratrol</i> $\rightarrow$ endothelial function via enhanced nitric oxide bioavailability	85.2	4.305

(a) Finite truth-value counts in Ω_{rw} .

(b) The two witness edges separate by similarity and joint support.

recovered as pullbacks of χ_{rw} along the points of Ω_{rw} selecting the desired truth value. In this way, the biomedical study is not merely a fuzzy SQL join: it is a concrete instance of predicate classification by a practical, finite, intuitionistic truth object.

Applying Topos Causal Models in the TCC Domain: Pullback and Method-Conflict Subobjects

We can also apply TCMs to lift the TCC economics corpus into a method-conditioned CSQL workflow. Here the ambient object is not a two-document

Table 46: Example query result: top corpus claims targeting *inflation* (ranked by paper support). In CSQL, such queries are expressible directly as SQL joins and filters over edges and *edge_support*.

Table 47: Two studies of the health benefits of red wine can be combined. The pushout decomposes into two glued witness rows plus source-specific rows from atlas A and atlas B.

Table 48: Soft-pullback witnesses that survive gluing across the two red-wine atlases.

Figure 55: Red-wine domain summary. Left: finite truth-value counts in the pushout provenance object. Right: the two cross-atlas witnesses in soft-pullback geometry.

pushout, but a pullback-style comparison between two method families over the shared (*src*, *dst*) claim interface. In the current demo, raw method strings are collapsed into *DID_FAMILY* (*DID*, *TWFE*, *Event Study*) and *IV_FAMILY* (*IV*, *2SLS*), producing a method-aligned corpus object with 265,656 support rows, 261,761 sign-split canonical edges, and 295,459 nodes.

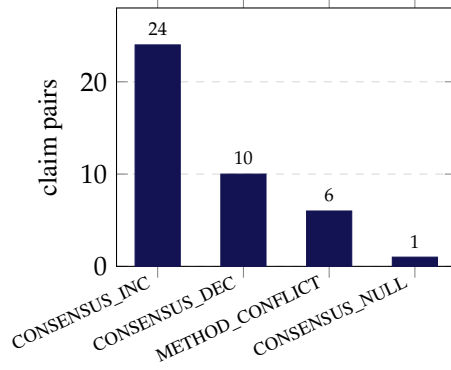
Source	Target	Sign	DID	IV	Masses
financial literacy	retirement planning	inc	2	5	7.8 / 8.2
education	wages	inc	2	3	4.8 / 4.2
income per capita	democracy	null	4	2	2.88 / 2.36
class size	student achievement	inc	1	2	3.0 / 2.0
education	earnings	inc	1	3	3.0 / 1.9

(a) Leading shared claims; the last column gives DID/IV mass.

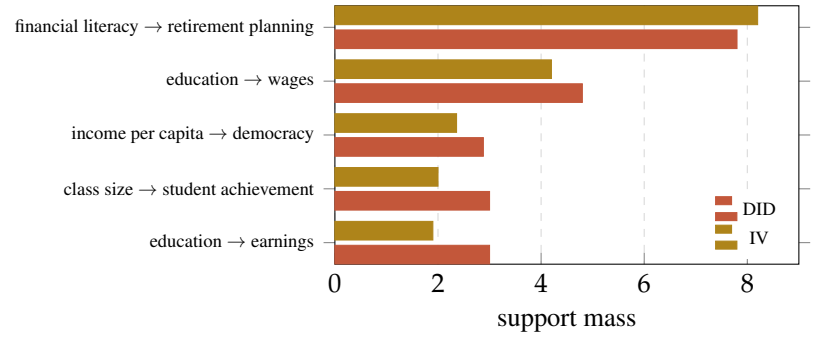
Ω value	Pairs
CONSENSUS_INC	24
CONSENSUS_DEC	10
METHOD_CONFLICT	6
CONSENSUS_NULL	1

(b) Method-conditioned truth values.

Table 49: TCC DID–IV method-pullback results: leading shared claims and their finite truth-value summary.



(a) Method-conditioned truth values in $\Omega_{\text{did/iv}}$.



(b) Top shared DID-vs-IV pullback claims by support mass.

PRACTICAL METHOD-LEVEL Ω . The TCC domain Topos Causal Model defines a second finite truth object

$$\Omega_{\text{did/iv}} = \{\text{CONSENSUS\_INC}, \text{CONSENSUS\_DEC}, \text{CONSENSUS\_NULL}, \text{METHOD\_CONFLICT}\},$$

and a characteristic map

$$\chi_{\text{did/iv}} : X_{\text{did/iv}} \rightarrow \Omega_{\text{did/iv}},$$

where $X_{\text{did/iv}}$ is the shared $\text{DID} \times \text{IV}$ pullback relation. A claim pair is classified as *CONSENSUS_INC*, *CONSENSUS_DEC*, or *CONSENSUS_NULL* only when the corresponding sign dominates within *both* method families by at least a factor of two over the competing signs; otherwise it is assigned to the *METHOD_CONFLICT* subobject. This makes disagreement between causal identification strategies itself queryable as a subobject. For example, the demo surfaces conflicts such as *democracy* → *economic growth*

Figure 56: TCC method-pullback summary: truth-value counts and support mass for the leading shared DID/IV claims.

and *exports* $\rightarrow$ *HIV incidence*, where overlapping method families support incompatible directional conclusions.

WHY THESE TWO STUDIES MATTER. Taken together, the red-wine and TCC demonstrations show that the topos-theoretic machinery defined earlier in the book can already be realized in a pragmatic form over causal databases. In both cases we choose an ambient CSQL object X , define predicates as row-level subobjects $S \hookrightarrow X$, construct a characteristic map $\chi : X \rightarrow \Omega$, and recover the classified rows by pullback along designated truth values. These finite Ω -objects are not yet the full subobject classifier of a completed topos of CSQL objects, but they make the central categorical pattern concrete: truth values are structured, non-Boolean, and operationally useful for gluing, filtering, and auditing causal claims.

CSQL over Other Domains

Domain	Nodes	Edges	Top Hub	Max Score
Human origins (The Washington Post)	501	65	bipedalism	1.462
Chocolate (The Washington Post)	171	31	regular physical exercise	21.271
Glyphosate (The New York Times)	652	92	glyphosate exposure	35.009
California drought (The New York Times)	441	66	exceptional wet period	2.209

The unexpected top hub in the chocolate atlas is an extraction-quality warning: a compiled claim database can contain off-topic discourse and should be audited before substantive use.

We have experimented with CSQL over a number of different domains. The domains analyzed are described below.

1. A recent newspaper article in The Washington Post on a potential causal link between dark chocolate and aging.³⁹ We used this domain as a running example throughout the paper to illustrate the type of causal analysis that DEMOCRITUS can perform.
2. The causal analysis was based on The Washington Post article on the origins of bipedal walking in human ancestors, dating back to 7 million years ago.⁴⁰
3. The NY Times published a recent article on the controversy surrounding the weedkiller Roundup, whose technical name is glyphosate⁴¹.
4. The NY Times published an article on the melting of glaciers in Antarctica⁴². The high-scoring model (top) contains a dense cluster of edges that are repeatedly supported across multiple discourse neighborhoods, including ice melt leading to freshwater influx, changes in ocean circulation, and rising global sea levels. These overlapping mechanisms yield strong model

Table 50: Summary statistics from four checked ATLAS artifacts. “Max Score” is the largest displayed `score_sum` and is not calibrated across domains.

³⁹ <https://www.washingtonpost.com/wellness/2025/12/26/dark-chocolate-health-benefits>

⁴⁰ <https://www.washingtonpost.com/science/2026/01/02/human-ancestor-biped/>

⁴¹ <https://www.nytimes.com/2026/01/02/climate/glyphosate-roundup-retracted-study.html?smid=nytcore-ios-share>.

⁴² <https://www.nytimes.com/2025/12/27/climate/antarctica-thwaites-glacier.html>

agreement and a high evaluation score. In contrast, the low-scoring model (bottom) contains fewer well-supported edges and weaker semantic coherence. Although such models are freely generated by the left adjoint, they fail to accumulate sufficient evidence during evaluation and are therefore suppressed before adjoint reconstruction.

Discussion and Implications

Relation to Knowledge Graphs and RAG

CSQL differs fundamentally from knowledge graphs and GraphRAG-style systems. Knowledge graphs store asserted facts; CSQL stores *evaluated causal hypotheses*. GraphRAG retrieves relevant nodes; CSQL computes credibility via aggregation across competing causal models.

Where knowledge graphs emphasize *what is stated*, CSQL emphasizes *what is structurally supported* by agreement across discourse.

Relation to Causal Inference

CSQL does not replace statistical causal inference. Instead, it complements it by operating in regimes where numerical data is unavailable but causal discourse is abundant. The output of CSQL can serve as a hypothesis generator, prior constructor, or explanatory scaffold for downstream causal analysis.

CSQL as a Causal Compiler

Viewed abstractly, CSQL is a compiler:

Documents $\longrightarrow$ Causal Models $\longrightarrow$ Relational Database.

The compilation target is a causally meaningful SQL schema equipped with a principled algebra of queries. This perspective connects CSQL to work on functorial databases and data migration, though users need not be aware of this machinery to use the system effectively.

Limitations and Future Work

While CSQL demonstrates that large document collections can be compiled into SQL-queryable causal databases, the current system has several important limitations. These limitations also suggest clear directions for future research and system development.

Limitations

DEPENDENCE ON DISCOURSE QUALITY. CSQL relies on large language models (LLMs) as discourse compilers that generate candidate causal statements. Although all downstream processing is deterministic, the quality and topical coherence of the resulting causal database depends on the quality of the extracted discourse. Noisy or off-domain discourse expansions can lead to diffuse causal atlases with weaker hub concentration and flatter score distributions. While the scoring and aggregation layers are designed to suppress unsupported claims, discourse quality remains a primary bottleneck.

NO GROUND-TRUTH CAUSAL VALIDATION. The causal relations represented in CSQL are inferred from language rather than validated against experimental or observational data. As a result, CSQL should be understood as capturing *discursive causal structure*—what is claimed, suggested, or argued in text—rather than verified causal ground truth. This distinction is fundamental: CSQL evaluates credibility under discourse agreement, not causal identifiability in the sense of statistical causal inference.

STATIC CAUSAL SNAPSHOTS. The current system constructs causal databases from static document collections. Temporal evolution of causal discourse—how claims change over time, how evidence accumulates or is retracted, and how controversies evolve—is not yet explicitly modeled. Each CSQL database represents a snapshot of discourse at the time the documents were processed.

LIMITED COMPOSITIONAL REASONING. Although CSQL supports multi-step causal queries via SQL joins (e.g., two-hop paths and hub analysis), it does not yet implement explicit causal algebraic operations such as intervention, counterfactual reasoning, or compositional diagram rewriting. These operations are implicit in the underlying theory but are not yet exposed at the database query layer.

SCALABILITY OF DISCOURSE COMPILATION. The most computationally expensive stage of the pipeline is discourse compilation via LLMs. While all subsequent steps scale efficiently and operate over Parquet and SQL engines, large-scale deployment over millions of documents will require additional optimizations, such as caching, incremental updates, or hybrid retrieval-generation strategies.

Future Work

TEMPORAL AND VERSIONED CAUSAL DATABASES. A natural extension of CSQL is to support time-indexed causal databases that track how causal claims evolve across document versions, publication dates, or evidence updates. This would enable queries such as “Which causal claims have

gained or lost support over time?” and would be particularly valuable for scientific, policy, and regulatory analysis. We explored fusing causal snapshots of companies from their 10-K filings in a previous chapter.

INTERVENTIONAL AND COUNTERFACTUAL QUERY LAYERS. Future versions of CSQL could expose higher-level causal operators on top of SQL, but doing so requires an external model that specifies mechanisms, intervention targets, and post-intervention propagation. Edge deletion or graph ablation alone is not a do-operator or a counterfactual semantics.

INTEGRATION WITH NUMERICAL DATA. Although CSQL currently operates on text-derived causal claims, it can naturally be extended to incorporate numerical datasets when available. For example, edges in the causal database could be linked to statistical models, regression results, or experimental evidence, enabling hybrid discourse–data causal analysis.

USER-GUIDED SCHEMA REFINEMENT. While CSQL deliberately avoids hand-designed ontologies, future work may explore lightweight human-in-the-loop mechanisms for refining concept normalization, merging equivalent nodes, or highlighting domain-relevant causal variables. Such guidance could improve interpretability without sacrificing generality.

DISTRIBUTED AND INCREMENTAL COMPILATION. To support large-scale deployments, future implementations will explore incremental atlas construction, distributed discourse compilation, and streaming updates to Parquet-backed causal tables. This would allow CSQL databases to be maintained continuously as new documents arrive.

CAUSAL CSQL AS A STANDARD INTERFACE. Finally, we envision CSQL as a candidate abstraction layer for causal querying over textual corpora, analogous to how SQL standardized access to relational data. A long-term goal is to define a portable causal query interface that can be embedded into analytics platforms, dashboards, and decision-support systems without requiring users to interact directly with causal modeling formalisms.

Suggested Reading

The CSQL system is described in [Mahadevan \[2026c\]](#). Functorial database schemas and canonical data migrations originate with [Spivak \[2012\]](#); [Fong and Spivak \[2018\]](#) gives a broader applied categorical account. For the interpretation of extracted economic claims, see [Garg and Fetzer \[2026\]](#).

Homotopy in Language and Causal Inference

Natural language can express one causal mechanism through many paraphrases, granularities, and discourse contexts. A causal-language system must identify variation that is semantically inessential without erasing differences in relation family, polarity, temporal scope, intervention type, or regime. This chapter develops the homotopy-localization layer used for that task in DEMOCRITUS [Mahadevan, 2026d].

The central problem is not merely that paraphrases should have similar embeddings or next-token distributions. It is that downstream causal artifacts should be invariant under declared normalization-preserving rewrites, while remaining sensitive to distinctions that alter causal meaning. For example,

“Digital investment drove direct-to-consumer growth,”

“Growth in direct-to-consumer channels was fueled by digital investment,” and

“Investment in digital platforms increased DTC sales”

may all support the same normalized causal claim. Treating them as unrelated fragments evidence; merging them without retaining source provenance overcounts support. Localization supplies a principled middle course.

This chapter develops a categorical homotopy perspective on this problem. Our starting point is the observation that there are at least three distinct levels at which equivalence may arise:

1. **Textual equivalence:** two statements are paraphrases of one another;
2. **Causal equivalence:** two statements induce the same normalized causal claim;
3. **Interventional equivalence:** two causal claims have the same interventional semantics relative to a chosen family of interventions.

The first level is the natural home of the original LLM homotopy problem. The second and third levels are what matter in practice for causal extraction, causal graph construction, and causal reasoning from language.

The central idea of the chapter is therefore a passage

textual mention $\longrightarrow$ normalized causal claim $\longrightarrow$ causal semantics,

and homotopy enters by identifying different linguistic realizations that should be treated as equivalent at one or more of these levels. This viewpoint

builds on our earlier work on homotopy in language models, but recasts it for causal discourse and separates the formal construction from its finite database approximation [Mahadevan, 2025e, 2026d].

Introduction

The problem of semantic variation is not merely cosmetic. If a system extracts causal knowledge from text, then paraphrases can lead to spurious duplication of edges, diluted evidence, and unstable downstream reasoning. In the simplest case, two equivalent statements may produce two nodes or two edges that should have been merged. In harder cases, multiple statements may agree at a coarse level but differ in polarity, temporal scope, or intervention type. This means that not every linguistic near-neighbor should be collapsed, and any notion of equivalence must be strong enough to identify genuine paraphrases while remaining sensitive to causal distinctions that matter.

The language-model case offers the simplest entry point. Given a phrase prefix x , an LLM defines a next-token distribution $p(\cdot \mid x)$. If x and x' are paraphrases, then one might hope that these predictive distributions are “the same” in an appropriate sense. Empirical methods such as k -NN language models smooth this behavior by interpolating learned probabilities with datastore-based nearest-neighbor estimates.⁴³ Our interest here is different: rather than proposing another smoothing heuristic, we want to identify the universal structure underlying the identification of paraphrases.

In causal extraction, the issue is deeper. A sentence fragment is not interesting merely as a predictor of the next token. It is interesting because it expresses a possible relation between latent or explicit variables, often with qualifiers involving time, polarity, regime, strength, uncertainty, and evidence source. Two linguistically different spans may therefore be equivalent as language, or equivalent as normalized causal claims, or equivalent only after passing to a still stronger notion of interventional meaning. The correct abstraction is not simply “homotopy in LLMs,” but *homotopy in language-induced causal representations*.

Why Homotopy Matters for DEMOCRITUS

The need for homotopy is not merely philosophical. It arises concretely in the operation of DEMOCRITUS, where the system must aggregate evidence from many textual formulations of a causal claim. Without a principled quotienting step, semantically equivalent claims are often stored as distinct edges, distinct causal statements, or distinct support sets. This produces three recurring pathologies.

First, *evidence fragmentation* occurs when multiple paraphrases of the

⁴³ Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models, 2020. URL <https://arxiv.org/abs/1911.00172>

same causal claim are treated as unrelated observations. This weakens support for the claim and distorts later scoring or ranking stages.

Second, *graph duplication* occurs when a causal graph contains several nearly identical edges that differ only because the underlying language differed. The resulting graph is not richer; it is merely noisier.

Third, *semantic instability* occurs when downstream procedures such as aggregation, contradiction detection, or intervention analysis become sensitive to linguistic accident rather than causal structure.

Homotopy addresses all three failures by introducing equivalence classes at the right level of abstraction. The raw textual mention is not the final object of interest. Rather, the goal is to pass from language to a normalized causal claim, and then from that claim to a stable causal semantics. This chapter develops the mathematical language for that passage.

From Language to Causal Content

We begin by separating syntax from causal semantics.

Definition 62. Let $\mathcal{S}_{\text{causal}}$ denote the *syntax category of causal mentions*. Its objects are textual spans, sentence fragments, or short passages extracted from a corpus that express putative causal content. A morphism

$$u \rightarrow v$$

in $\mathcal{S}_{\text{causal}}$ represents a semantics-preserving rewrite, paraphrase map, normalization step, or discourse-level transformation from the mention u to the mention v .

The objects of $\mathcal{S}_{\text{causal}}$ are not yet causal variables or structural equations. They are linguistic artifacts: the things a system actually sees in text.

Definition 63. Let $\mathcal{C}_{\text{causal}}$ denote the *causal semantics category*. An object of $\mathcal{C}_{\text{causal}}$ is a normalized causal claim

$$c = (X, Y, r, s, \tau, \kappa, q),$$

where

- X is a cause variable or structured cause expression,
- Y is an effect variable or structured effect expression,
- r is a relation type (such as causes, promotes, inhibits, correlates-with, enables),
- s is polarity or signed direction,
- τ is temporal scope,
- κ is regime, context, or conditioning environment,

- q records qualifiers such as hedging, evidence source, strength, or confidence.

Morphisms in $\mathcal{C}_{\text{causal}}$ represent structure-preserving maps between normalized causal claims, such as ontological refinement, context-preserving identification, or admissible weakening/strengthening of qualifiers.

The point of $\mathcal{C}_{\text{causal}}$ is that it is already on the causal side of the syntax/semantics divide. A sentence such as “Investment in digital platforms increased direct sales” is not stored simply as raw text, but as a structured claim with explicit slots that can later be embedded into a causal graph or Markov-category semantics.

Definition 64. A *causal extraction functor* is a functor

$$E : \mathcal{S}_{\text{causal}} \rightarrow \mathcal{C}_{\text{causal}}$$

that maps each textual mention to its normalized causal claim.

In practice, E is implemented by a pipeline that performs entity normalization, relation identification, polarity detection, temporal parsing, qualifier extraction, and confidence scoring. The point of introducing it as a functor is that it allows us to define weak equivalence in a way that is adapted to causal extraction rather than raw sentence similarity.

Definition 65. A morphism $f : u \rightarrow v$ in $\mathcal{S}_{\text{causal}}$ is a **weak equivalence** if $E(f)$ is an isomorphism in $\mathcal{C}_{\text{causal}}$. Equivalently, u and v are weakly equivalent when they induce the same normalized causal content.

Let W_E contain all objects of $\mathcal{S}_{\text{causal}}$ and exactly those morphisms sent by E to isomorphisms.

Proposition 3 (Normalization-induced relative category). *The pair $(\mathcal{S}_{\text{causal}}, W_E)$ is a relative category, and its weak equivalences satisfy the two-out-of-three property.*

Proof. Functoriality sends identities to identities and composites of morphisms in W_E to composites of isomorphisms. Hence W_E is a wide subcategory. If any two of $E(f)$, $E(g)$, and $E(gf) = E(g)E(f)$ are isomorphisms, then so is the third, which proves two-out-of-three. $\square$

Thus, weak equivalence is not merely lexical similarity. Two statements are weakly equivalent only when the causal extraction functor cannot distinguish them at the level of normalized causal claims.

Example 15. *The three textual mentions*

“Supply chain disruptions hurt margins,”
 “Margins were pressured by supply chain issues,” and
 “Logistics bottlenecks reduced profitability”

may all map under E to the same normalized claim

$(\text{supply\_chain\_disruption}, \text{margin}, \text{inhibits}, -, \tau, \kappa, q).$

In that case they lie in the same weak-equivalence class in $\mathcal{S}_{\text{causal}}$.

Localization and the Homotopy Category

The next step is to formally invert weak equivalences.

Definition 66. Let W_E denote the class of weak equivalences in $\mathcal{S}_{\text{causal}}$ induced by the extraction functor E . When the localization exists, the **homotopy category of causal mentions** is the category

$$\text{Ho}(\mathcal{S}_{\text{causal}}) := \mathcal{S}_{\text{causal}}[W_E^{-1}],$$

obtained by formally inverting the morphisms in W_E .

Write

$$Q : \mathcal{S}_{\text{causal}} \longrightarrow \mathcal{S}_{\text{causal}}[W_E^{-1}]$$

for the localization functor. Because E sends every member of W_E to an isomorphism, the universal property of localization gives a functor

$$\bar{E} : \mathcal{S}_{\text{causal}}[W_E^{-1}] \longrightarrow \mathcal{C}_{\text{causal}}, \quad \bar{E}Q = E,$$

unique up to the canonical uniqueness of the factorization. Thus normalization is not merely followed by an informal clustering step: it descends through the category in which admissible rewrites have been inverted.

This is the correct causal analogue of the category of fractions. Its conceptual meaning is simple: in the homotopy category, all textual realizations of the same normalized causal claim become isomorphic.

Proposition 4. Suppose $u, v \in \mathcal{S}_{\text{causal}}$ are connected by a zig-zag of weak equivalences in W_E . Then u and v become isomorphic in $\text{Ho}(\mathcal{S}_{\text{causal}})$.

Proof. By construction, every weak equivalence in W_E is inverted in the localization $\mathcal{S}_{\text{causal}}[W_E^{-1}]$. Hence any zig-zag of such morphisms becomes an isomorphism in the localized category. $\square$

This observation specifies what DEMOCRITUS needs, but it must not be read as saying that the implementation computes a theorem-prover-certified localization. The canonical claim classes stored in CSQL are a finite computational quotient approximating this construction. They preserve the normalization fields and provenance needed to audit which mentions were merged and which distinctions blocked a merge.

Remark 15. Localization should be understood here as the primary construction. It is the mathematically correct way to express the statement “different phrasings that induce the same causal content should be identified.” This is a stronger and cleaner claim than merely saying that similar sentences ought to have similar embeddings or similar next-token distributions.

Homotopical Repair of a Causal-Discourse Sketch

Localization says which variations may be ignored; it does not yet say which local claims should compose. Let

$$S_{\text{Dem}} = (S, \mathcal{D}, \mathcal{L}, \mathcal{K})$$

be a finite sketch whose generators represent mentions, normalized claims, documents, regimes, overlaps, and corpus-level assemblies. The path equations $\mathcal{D}$ express agreement among normalization and refinement routes; the distinguished cones $\mathcal{L}$ describe document- and regime-local views over common interfaces; and the distinguished cocones $\mathcal{K}$ describe provenance-preserving assemblies. If $J = \text{Path}(S)$ and $q_{\mathcal{D}} : J \rightarrow J/\sim_{\mathcal{D}}$ is the quotient by the declared path equations, a corpus realization is a diagram

$$D : J \longrightarrow \mathcal{S}_{\text{causal}}.$$

Strict compositionality asks D to factor through $q_{\mathcal{D}}$ while realizing the distinguished cones and cocones. Homotopical compositionality asks instead for

$$\tilde{D} : J/\sim_{\mathcal{D}} \longrightarrow \mathcal{S}_{\text{causal}}[W_E^{-1}] \quad \text{and} \quad QD \cong \tilde{D}q_{\mathcal{D}}.$$

A homotopy repair replaces or augments the observed diagram by weakly equivalent data for which this factorization is inhabited. The obstruction is therefore a failed structural factorization after admissible linguistic variation has been inverted, not a scalar distance between sentence embeddings.

This is a base-level repair problem. The current system does not define a tangent functor on causal discourse and does not compute an infinitesimal factorization problem for TD . Homotopy localization and infinitesimal non-compositionality are related layers, but they are not the same algorithm.

LLMs as a Special Case

The original LLM homotopy problem is recovered as a special case of the above framework.

Definition 67. Let $\mathcal{L}$ denote a $[0, 1]$ -enriched syntax category of language prefixes. Its objects are strings or token prefixes, and its enriched hom-objects $\mathcal{L}(x, y) \in [0, 1]$ measure the strength with which y is an admissible continuation or semantics-preserving extension of x .

A semantic category may then be constructed using enriched copresheaves.

Definition 68. The *semantic category* associated with $\mathcal{L}$ is the $[0, 1]$ -enriched functor category

$$\hat{\mathcal{L}} := [0, 1]^{\mathcal{L}}.$$

Its objects are $[0, 1]$ -enriched copresheaves $F : \mathcal{L} \rightarrow [0, 1]$, which may be interpreted as semantic neighborhoods or completion profiles.

In the pure LLM case, the extraction functor E above may be replaced by a paraphrase normalization functor from textual prefixes to semantic profiles. The point of the present chapter, however, is that the LLM case is not the end of the story. It is the simplest instance of a much broader pipeline in which language is used to recover causal structure.

Remark 16. *Empirical methods such as k -NN language models may be viewed as practical approximations to the idea that semantically neighboring expressions ought to share predictive behavior. Homotopy theory seeks something more abstract: a principled account of when superficially different expressions should be identified by a universal construction rather than by a heuristic smoothing rule.*

Causal Semantics and Markov Categories

The normalized claim category $\mathcal{C}_{\text{causal}}$ is still not the final semantic layer. To reason causally, we want to interpret normalized claims inside a category that supports stochastic maps, copying, deleting, and interventions. Markov categories provide exactly this structure.

Definition 69. *Let $\mathcal{M}$ be a Markov category representing the stochastic semantics of a causal domain. A **semantic realization functor** is a functor*

$$\llbracket - \rrbracket : \mathcal{C}_{\text{causal}} \rightarrow \mathcal{M}$$

that maps each normalized causal claim c to a stochastic morphism or structured morphism in $\mathcal{M}$.

Thus the full semantic passage is

$$\mathcal{S}_{\text{causal}} \xrightarrow{E} \mathcal{C}_{\text{causal}} \xrightarrow{\llbracket - \rrbracket} \mathcal{M}.$$

This architecture separates three sources of variation:

- purely linguistic variation,
- differences in normalized causal form,
- genuine differences in stochastic or interventional semantics.

Definition 70. *Let $\mathcal{I}$ be a chosen family of interventions appropriate to the domain. Two normalized causal claims $c, c' \in \mathcal{C}_{\text{causal}}$ are **interventionally equivalent**, written*

$$c \simeq_{\text{int}} c',$$

if their realizations agree on all interventions in $\mathcal{I}$.

The precise form of “agree” depends on the chosen semantics. In a structural-causal setting it may mean equality of interventional distributions.

In a Markov-category setting it may mean equality of the corresponding stochastic morphisms after composition with admissible intervention maps. In a regime-based or j -stable setting, it may mean agreement across a chosen cover of comparable regimes.

Remark 17. *This distinction matters in practice. Statements such as “ X causes Y ,” “ X contributes to Y ,” and “ X is associated with Y ” may be linguistically close, and may even project to nearby embeddings, but they are not generally interventionally equivalent. Homotopy in causal inference must therefore be stronger than paraphrase matching and weaker than indiscriminate semantic smoothing.*

Paraphrase Groupoids and Simplicial Structure

Once weak equivalences have been identified, it is natural to organize equivalent textual realizations into groupoids and simplicial objects.

Definition 71. *For a normalized causal claim $c \in \mathcal{C}_{\text{causal}}$, the **paraphrase groupoid** $\mathcal{G}_c$ is the groupoid whose objects are textual mentions $u \in \mathcal{S}_{\text{causal}}$ satisfying $E(u) \cong c$, and whose morphisms are invertible paraphrase maps between them.*

The groupoid $\mathcal{G}_c$ captures all known textual realizations of the same normalized claim.

Definition 72. *The **nerve** $N(\mathcal{G}_c)$ of the paraphrase groupoid $\mathcal{G}_c$ is the simplicial set whose n -simplices are strings of n composable morphisms in $\mathcal{G}_c$. Its geometric realization is the classifying space $B\mathcal{G}_c$.*

This construction has a clear interpretation. The vertices of $N(\mathcal{G}_c)$ are textual realizations of the claim c . The 1-simplices are pairwise paraphrase identifications. The 2-simplices represent coherent triples of paraphrases. Higher simplices encode higher-order compatibility among multiple realizations.

The nerve of a genuine groupoid is a Kan complex: its horns already have fillers. It is therefore an idealized coherently completed object, not the place where missing fillers diagnose extraction failure. The implemented diagnostic must live in a different finite object.

Definition 73. *For a normalized claim c , the **observed compatibility complex** K_c^{obs} has extracted mentions eligible to represent c as vertices. Its edges are pairwise compatibility decisions actually recorded by the system, and an n -simplex is present only when the corresponding $(n + 1)$ -way comparison has been explicitly certified as jointly compatible.*

Face closure requires every verified multiway comparison to restrict to verified lower-order comparisons. A missing candidate simplex in K_c^{obs} may

consequently record absent evidence, an unperformed test, or a genuine failure of observed coherence. Completing this finite complex toward the nerve of a paraphrase groupoid is a mathematical ideal; the implemented system reports finite compatibility decisions rather than performing fibrant replacement.

Weighted and Fuzzy Simplicial Structure

Confidence scores can be incorporated directly into this picture.

Definition 74. A *weighted paraphrase complex* for a normalized claim c is a simplicial set X_c together with a weight map

$$w : X_c \rightarrow [0, 1]$$

assigning a confidence to each simplex, subject to the condition that the weight of a simplex is bounded above by the minimum of the weights of its faces.

The weight $w(\sigma)$ may incorporate extraction confidence, agreement across documents, ontology match quality, or inter-annotator reliability. This is a possible extension of the observed complex; it should not be confused with an implemented or externally calibrated confidence model in the present system.

Homotopy, Obstructions, and Aggregation in DEMOCRITUS

We can now state more clearly what role homotopy plays in practice.

Evidence pooling

When multiple textual mentions lie in the same weak-equivalence class, their support can be aggregated rather than scattered across duplicate graph edges, provided that source provenance is retained. Localization provides the formal justification for the identification; it does not make repeated paraphrases independent evidence.

Graph consolidation

A causal graph constructed from text should be built from normalized homotopy classes of mentions, not from raw mentions themselves. This avoids the pathology in which the same causal relation is counted multiple times under slightly different phrasings.

Ambiguity detection

If a family of mentions fails to assemble into a coherent paraphrase groupoid or observed compatibility complex, this signals ambiguity. Such ambiguity

may come from true semantic drift, regime dependence, temporal mismatch, clause-tail segmentation, an unperformed comparison, or genuinely competing causal interpretations. The implementation accordingly distinguishes coherent, partially glued, and disconnected families rather than assigning a single similarity score.

Contradiction and polarity sensitivity

Homotopy does not mean indiscriminate merging. Statements that differ in polarity, intervention type, or temporal scope should not be collapsed merely because they share lexical material. The role of the normalized causal semantics category $\mathcal{C}_{\text{causal}}$ is precisely to ensure that these distinctions are visible before localization is performed.

A motivating example

Consider the following statements extracted from annual filings:

“We continue to invest in digital platforms to expand direct channels.”

“Digital investment is accelerating growth in our direct-to-consumer business.”

“Our DTC expansion has been enabled by continued spending on digital capabilities.”

A naive system may produce three distinct edges. The homotopy perspective instead treats these as three points in $\mathcal{S}_{\text{causal}}$ that map under E to the same normalized claim in $\mathcal{C}_{\text{causal}}$, and hence to the same equivalence class in $\text{Ho}(\mathcal{S}_{\text{causal}})$. The resulting graph contains a single consolidated claim with aggregated support and preserved source provenance.

Empirical Boundary of the Current System

The extraction benchmark reported in the causal-claims chapter calibrates the upstream interface against the public UniCausal models on a frozen set of 404 grouped AltLex passages. UniCausal obtains classification F1 (59.2%) and macro span F1 (43.6%), compared with (57.1%) and (39.1%) for the restricted DEMOCRITUS adapter. The adapter has higher recall ((80.9%) versus (68.7%)) but lower precision ((44.1%) versus (52.0%)). This is a useful external comparison, but it does not evaluate homotopy localization: the benchmark deliberately bypasses retrieval, normalization, localization, and gluing.

The post-extraction evidence in the journal study has a different character. Saved CSQL artifacts expose normalized claim classes, support counts, regime projections, gluing labels, and source provenance. Case studies show both stable repeated mechanisms and corpora that remain collections of local patches rather than one global causal account. These artifacts support claims of inspectability and diagnostic usefulness. They do not establish that the

extracted claims are causally true, and no independent expert panel has yet annotated every merge, non-merge, or gluing decision.

Admission contract: Three validation questions

1. **Extraction:** Did the system find the annotated causal passage and its cause and effect spans? The UniCausal comparison addresses this layer.
2. **Localization:** Did normalization merge paraphrases without collapsing polarity, time, or regime distinctions? Current evidence comes from saved-artifact audits and representative cases.
3. **Causal grounding:** Do localized discourse claims correspond to identified causal mechanisms under stated assumptions? Text alone does not answer this question.

On Model Structures

The simplicial language above naturally invites the use of model categories, but one must be careful about where this structure lives.

The raw syntax category of textual mentions is not, by itself, automatically a model category. Nor should one claim that every LLM category or causal mention category directly carries a Quillen model structure. The correct statement is more modest and more precise: once we pass to appropriate simplicial or groupoidal constructions — for example, nerves of paraphrase groupoids or simplicial sets built from localized syntax categories — standard homotopical machinery becomes available. It is at this level that one can invoke familiar model structures on simplicial sets.⁴⁴

⁴⁴ Daniel G. Quillen. *Homotopical algebra*. Springer, 1967

This is enough for our purposes. The central construction in the present chapter is localization at weak equivalences, not a blanket assertion that raw language categories are already model categories. Simplicial and model-categorical tools then enter as a powerful secondary layer for analyzing the geometry of the resulting equivalence classes.

Outlook: Homology and Higher Invariants

Localization is the first construction needed here, but it need not be the last. Once causal mentions have been assembled into independently validated simplicial objects, one may define chain complexes, homology groups, and obstruction measures. Whether those higher invariants improve the diagnosis of unresolved gaps, inconsistency cycles, or regime-sensitive failures is an open empirical question.

From the perspective of DEMOCRITUS, this suggests a hierarchy of analysis:

1. extract causal mentions from text;
2. normalize them into structured causal claims;
3. localize by weak equivalences to collapse paraphrases;
4. assemble the resulting families into weighted simplicial objects;
5. compute homotopic and homological invariants of the resulting structures.

The implemented system approximates the first three steps with normalized database keys and stored comparison decisions. The latter two require a more complete compatibility benchmark before their empirical value can be assessed.

Bridge to Markov Categories, j -Stable Semantics, and Topos Causality

The constructions introduced in this chapter are preparatory for several later themes in the book. Their purpose is not exhausted by paraphrase handling. Rather, they supply the quotient structure needed to make later causal semantics well posed.

First, the passage from textual mentions to normalized causal claims is what allows the Markov-category semantics developed earlier in the book to be applied coherently. A Markov morphism should correspond to a stable causal content, not to an arbitrary surface form in language. Homotopy therefore plays a bookkeeping role that is mathematically essential: it determines when two linguistic realizations should induce the same stochastic arrow.

Second, the distinction between causal equivalence and interventional equivalence prepares the ground for the later treatment of interventions. Two statements may be paraphrastic and hence textually homotopic, yet fail to be equivalent once one accounts for regime, polarity, conditioning context, or intervention type. This is precisely the setting in which the j -stable perspective becomes important. A family of claims may glue at the linguistic level while failing to glue over a cover of interventional regimes. Thus homotopy at the language level is only the first stage of a more refined gluing problem.

Third, the simplicial and groupoidal constructions introduced here point naturally toward the topos-theoretic perspective developed later. Paraphrase classes, weighted simplicial assemblies, and coherence conditions all suggest that causal meaning extracted from language should be studied not merely as a set of isolated claims, but as a structured sheaf-like object over contexts, regimes, and evidential viewpoints. In that later setting, failures of semantic agreement appear not simply as noisy language, but as failures of descent, failures of compatibility, or obstructions to gluing.

From this viewpoint, the chapter can be read as establishing the following progression:

language $\longrightarrow$ homotopy class of mentions $\longrightarrow$ normalized causal claim
 $\longrightarrow$ Markov/interventional semantics $\longrightarrow$ topos-theoretic gluing over regimes.

This progression is one of the main conceptual threads of the book. Homotopy enters first because language varies. It remains indispensable later because causal semantics must be invariant under that variation.

Summary

This chapter reframed the original problem of homotopy in large language models as a broader problem of *homotopy in language and causal inference*. The practical challenge is not merely that paraphrases should produce similar next-token distributions. It is that language induces causal claims, and superficially different formulations often express the same causal content. To handle this, we introduced three levels of equivalence: textual equivalence, causal equivalence, and interventional equivalence.

We defined a syntax category $\mathcal{S}_{\text{causal}}$ of causal mentions, a causal semantics category $\mathcal{C}_{\text{causal}}$ of normalized claims, and a causal extraction functor

$$E : \mathcal{S}_{\text{causal}} \rightarrow \mathcal{C}_{\text{causal}}.$$

Weak equivalences in the syntax category were defined by those morphisms that become isomorphisms after applying E , and the resulting localization

$$\text{Ho}(\mathcal{S}_{\text{causal}}) = \mathcal{S}_{\text{causal}}[W_E^{-1}]$$

provides the correct homotopy category for collapsing paraphrase-induced duplication in causal extraction.

We then connected normalized causal claims to stochastic semantics through a Markov-category realization functor, thereby distinguishing paraphrase-level equivalence from stronger notions of interventional equivalence. Finally, we separated the nerve of an idealized paraphrase groupoid from the observed compatibility complex used for finite diagnostics. This distinction matters: groupoid nerves already satisfy horn filling, whereas missing or uncertified multiway comparisons belong to the observed complex.

The central lesson is that homotopy is not merely decorative terminology. For systems such as DEMOCRITUS, it supplies a useful organizing specification for invariance under declared rewrites. Without it, causal evidence fragments across linguistic variants. With it, one obtains a principled specification for passing from raw text to normalized causal claims while preserving distinctions that matter for causal interpretation. The current DEMOCRITUS system realizes this specification approximately through localized claim

classes, regime-sensitive gluing labels, observed compatibility records, and provenance-preserving artifacts. These objects organize causal discourse; they do not certify causal truth.

Suggested Reading

The classical foundations are Quillen’s *Homotopical Algebra* [Quillen, 1967], supplemented by May [1992] for simplicial objects. The application to linguistic equivalence is developed in Mahadevan [2025e]; nearest-neighbor language models provide a useful operational comparison [Khandelwal et al., 2020]. The normalization-induced relative category, homotopical sketch repair, observed compatibility complex, and empirical UniCausal comparison are developed in Mahadevan [2026d]; see Tan et al. [2023] for the benchmark and public causal-text-mining models.

Model Categories for Causality and Language

Earlier chapters introduced categorical approaches to language models, causal extraction, and stochastic semantics. A recurring practical difficulty in all of these settings is that natural language expresses the same content in many different forms. For large language models, this appears as the paraphrase problem: expressions such as “Charles Darwin wrote” and “Charles Darwin is the author of” ought to support the same continuation behavior, yet often do not. For causal extraction systems such as DEMOCRITUS, the issue is even more severe: distinct textual mentions of the same causal claim may be treated as separate edges, separate support sets, or separate causal statements.

This chapter introduces a concise axiomatic framework for this problem using homotopy theory in the sense of model categories. The central idea is simple: we should not reason directly with raw linguistic surface forms, but rather with objects defined only up to a suitable notion of weak equivalence. Model categories provide one possible language for doing this. They distinguish three classes of morphisms — weak equivalences, fibrations, and cofibrations — and organize them through lifting and factorization principles.

Our goal in this chapter is not to place every language or causal category directly inside a model structure. Rather, the aim is to show how model-categorical language suggests an abstract framework for paraphrase invariance, causal normalization, and homotopy classes of text-induced causal claims. The chapter is therefore organized around three ideas:

language $\longrightarrow$ normalized causal content $\longrightarrow$ homotopy classes under weak equivalence.

Why Language and Causality Need Weak Equivalences

The basic problem is that syntactic difference should not automatically imply semantic difference. In ordinary language understanding, multiple phrases may express the same proposition. In causal inference from text, multiple sentences may support the same causal relation. If a system fails to identify these equivalences, then evidence fragments.

For example, consider the three statements

“Digital investment drove direct-to-consumer growth,”

“Growth in direct-to-consumer channels was fueled by digital investment,”

“Investment in digital platforms increased DTC sales.”

A naive extraction system may produce three distinct causal edges. A better system should recognize that these are three linguistic realizations of the same underlying causal claim.

This motivates the introduction of an explicit equivalence relation. But in category theory, equivalence is often better handled not as a mere relation on objects, but as a class of morphisms to be inverted. This leads directly to the language of weak equivalences.

Syntax, Causal Semantics, and Extraction

We begin by separating textual syntax from normalized causal content.

Definition 75. Let $\mathcal{S}_{\text{causal}}$ denote the *syntax category of causal mentions*. Its objects are textual spans, sentence fragments, or short passages expressing putative causal content. A morphism

$$u \rightarrow v$$

in $\mathcal{S}_{\text{causal}}$ represents a semantics-preserving rewrite, paraphrase map, normalization step, or discourse-level reformulation.

Thus $\mathcal{S}_{\text{causal}}$ is still a language-facing category: its objects are the things a system reads in text.

Definition 76. Let $\mathcal{C}_{\text{causal}}$ denote the *causal semantics category*. An object of $\mathcal{C}_{\text{causal}}$ is a normalized causal claim

$$c = (X, Y, r, s, \tau, \kappa, q),$$

where X is a cause variable, Y is an effect variable, r is a relation type, s is polarity, τ is temporal scope, κ is regime or context, and q records qualifiers such as hedging, confidence, or evidence source. Morphisms in $\mathcal{C}_{\text{causal}}$ represent structure-preserving maps between normalized claims.

The important point is that $\mathcal{C}_{\text{causal}}$ is already on the causal side of the syntax/semantics divide. A textual expression has been stripped of linguistic accident and mapped to a structured causal object.

Definition 77. A *causal extraction functor* is a functor

$$E : \mathcal{S}_{\text{causal}} \rightarrow \mathcal{C}_{\text{causal}}$$

mapping each textual mention to its normalized causal claim.

This functor is the mathematical abstraction of what systems such as DEMOCRITUS attempt operationally: entity normalization, relation extraction, polarity analysis, temporal parsing, and confidence assignment.

Definition 78. A morphism $f : u \rightarrow v$ in $\mathcal{S}_{\text{causal}}$ is called a **weak equivalence** if $E(f)$ is an isomorphism in $\mathcal{C}_{\text{causal}}$. We denote the class of such morphisms by W_E .

So weak equivalences are precisely those rewrites that preserve normalized causal content.

Example 16. Suppose the three statements above all map under E to the same normalized claim

$$(digital\_investment, dtc\_growth, promotes, +, \tau, \kappa, q).$$

Then the morphisms connecting these textual mentions belong to W_E . They differ linguistically, but not causally.

Localization and Homotopy Categories

Once weak equivalences have been identified, the next step is to formally invert them.

Definition 79. When the localization exists, the **homotopy category of causal mentions** is

$$\text{Ho}(\mathcal{S}_{\text{causal}}) := \mathcal{S}_{\text{causal}}[W_E^{-1}],$$

the category obtained from $\mathcal{S}_{\text{causal}}$ by formally inverting all weak equivalences in W_E .

This localized category expresses the core semantic idea of the chapter: textual mentions that differ only by causally inessential reformulation become isomorphic after localization.

Proposition 5. If $u, v \in \mathcal{S}_{\text{causal}}$ are connected by a zig-zag of weak equivalences in W_E , then u and v become isomorphic in $\text{Ho}(\mathcal{S}_{\text{causal}})$.

Proof. By definition of the localization, every morphism in W_E becomes invertible. Hence every zig-zag of weak equivalences becomes an isomorphism. $\square$

The significance for causal extraction is immediate. Downstream reasoning should not operate on raw mentions but on their images in $\text{Ho}(\mathcal{S}_{\text{causal}})$, where paraphrase-induced duplication has already been quotiented out.

Remark 18. This localization step should be regarded as primary. It is the mathematically clean version of the operational instruction “merge paraphrases that express the same causal claim.”

Lifting Problems

To go beyond localization and toward axiomatic homotopy theory, we need the language of lifting problems.

Definition 80. Let $\mathcal{C}$ be a category. A **lifting problem** in $\mathcal{C}$ is a commutative square

$$\begin{array}{ccc} A & \xrightarrow{\mu} & X \\ f \downarrow & & \downarrow p \\ B & \xrightarrow{\nu} & Y \end{array}$$

A **solution** is a morphism $h : B \rightarrow X$ such that $h \circ f = \mu$ and $p \circ h = \nu$.

Definition 81. We say that f has the **left lifting property** with respect to p , or that p has the **right lifting property** with respect to f , if every lifting problem of the above form admits a solution.

This simple notion turns out to be one of the central organizing ideas of axiomatic homotopy theory. In the present setting, lifting may be interpreted as the ability to extend a partial semantic match, a partial causal normalization, or a partially specified paraphrase identification to a full one.

Example 17. Suppose A is a partial textual mention, B is a fuller reformulation, X is a partially normalized causal representation, and Y is a coarse causal target. A lift $h : B \rightarrow X$ expresses the fact that the richer mention B can be made compatible with the more refined semantic representation X .

Lifting is therefore a formal language for discussing whether local pieces of syntactic or causal information can be glued into coherent global structure.

Simplicial Models of Paraphrase Coherence

To study higher-order coherence, it is useful to organize paraphrases and causal normalizations simplicially.

Definition 82. A **simplicial set** is a contravariant functor

$$X : \Delta^{op} \rightarrow \mathbf{Set},$$

where Δ is the simplicial category of finite ordinals and order-preserving maps.

For our purposes, the main intuition is straightforward. Vertices represent textual or causal objects. Edges represent pairwise identifications or rewrites. Triangles represent coherent triples of identifications. Higher simplices encode higher-order compatibility.

Definition 83. For a normalized causal claim c , the **paraphrase groupoid** $\mathcal{G}_c$ has as objects all mentions $u \in \mathcal{S}_{\text{causal}}$ with $E(u) \cong c$, and as morphisms all invertible paraphrase maps between them. Its nerve $N(\mathcal{G}_c)$ is the simplicial set whose n -simplices are chains of n composable morphisms in $\mathcal{G}_c$.

The nerve construction is important because it converts equivalence data into a combinatorial topological object.

Definition 84. A *horn* $\Lambda_i^n \subseteq \Delta^n$ is the simplicial subset obtained by removing the interior of the n -simplex together with one of its faces. A simplicial set is a **Kan complex** if every horn can be filled.

In our setting, horn filling has a natural interpretation: if several pairwise paraphrase relations are known, a horn filler corresponds to a coherent higher-order identification that completes the pattern.

Remark 19. We do not claim here that every syntax category or every causal extraction system automatically gives rise to a Kan complex. Rather, simplicial constructions such as nerves of paraphrase groupoids provide a natural arena in which standard homotopical machinery can be applied.

Model Categories

We now arrive at the axiomatic framework itself.

Definition 85. A *model category* $\mathcal{M}$ is a category equipped with three classes of morphisms:

- *weak equivalences*,
- *cofibrations*,
- *fibrations*,

subject to the usual axioms:

1. $\mathcal{M}$ has all finite limits and colimits (many standard examples are complete and cocomplete);
2. weak equivalences satisfy the two-out-of-three property;
3. weak equivalences, cofibrations, and fibrations are closed under retracts;
4. cofibrations lift against acyclic fibrations, and acyclic cofibrations lift against fibrations;
5. every morphism factors both as a cofibration followed by an acyclic fibration, and as an acyclic cofibration followed by a fibration:

$$A \xrightarrow{\text{cofibration}} Z \xrightarrow{\text{acyclic fibration}} B \quad \text{and} \quad A \xrightarrow{\text{acyclic cofibration}} Z' \xrightarrow{\text{fibration}} B.$$

The importance of this definition is that it provides an abstract calculus of equivalence without requiring one to work directly in topological spaces.

In the present chapter, the model-category axioms should be read interpretively. No model structure on raw causal mentions is constructed or verified here.

- **Weak equivalences** represent semantics-preserving rewrites: distinct textual or causal presentations that induce the same normalized content.

- **Cofibrations** represent controlled extensions of partial structure: for example, enlarging a partial mention, adding qualifiers, or embedding a partial causal pattern into a larger one.
- **Fibrations** represent maps along which compatible refinements can be lifted: they encode semantic stability under refinement.

The lifting axioms then say that suitably compatible partial structures can be extended, while the factorization axioms say that every map can be decomposed into a “structured inclusion plus equivalence” or an “equivalence plus stable projection.”

Remark 20. *For language and causal inference, the model structure is typically not placed directly on raw text. Instead, one may pass to a specified simplicial, groupoidal, or localized construction for which a model structure is already known, or prove the model axioms for a new construction. Merely naming weak equivalences, fibrations, and cofibrations does not supply the lifting and factorization axioms.*

Axiomatic Homotopy for LLMs and Causal Inference

We can now state the book-level lesson in a compact form.

In language modeling, the raw objects are textual prefixes or fragments. In causal inference from text, the raw objects are mentions of possible causal claims. In both cases, surface form is too fine-grained. A useful theory must identify presentations that are equivalent up to meaning.

Model-categorical language provides a candidate abstraction:

- weak equivalences identify semantics-preserving variation,
- homotopy categories localize away from such variation,
- fibrations and cofibrations control how partial descriptions extend and lift,
- simplicial constructions encode higher-order coherence.

This perspective applies to LLMs and causal inference in slightly different ways.

For LLMs, one may regard paraphrastic variation as defining weak equivalences between prefixes, contexts, or semantic neighborhoods. For causal inference, the more important step is the extraction functor

$$E : \mathcal{S}_{\text{causal}} \rightarrow \mathcal{C}_{\text{causal}},$$

which defines equivalence not by lexical proximity but by preservation of normalized causal content.

The causal setting is stricter. Two phrases may be near paraphrases linguistically while differing in polarity, temporality, modality, or intervention type. Thus the extraction functor is essential: it prevents homotopy from collapsing distinctions that are causally meaningful.

Bridge to Markov Semantics and Interventions

This chapter provides an axiomatic layer beneath several later themes in the book.

First, once normalized causal claims have been identified up to weak equivalence, they may be interpreted in Markov categories or related stochastic semantics. Homotopy is therefore a stabilization layer between language and probabilistic causal meaning.

Second, the distinction between textual equivalence and stronger causal or interventional equivalence prepares the ground for later chapters on interventions. Two claims may be weakly equivalent as language while failing to be equivalent once regime, polarity, or intervention type is taken into account.

Third, the simplicial viewpoint anticipates later discussions of gluing, compatibility, and obstruction. When paraphrases fail to assemble coherently, the problem is no longer merely linguistic noise; it becomes a failure of higher-order compatibility.

Thus the present chapter establishes the progression

language $\longrightarrow$ weak equivalence class $\longrightarrow$ normalized causal claim $\longrightarrow$ stochastic/interventional semantics.

Summary

This chapter recast the problem of semantic variation in language and causal inference in the language of axiomatic homotopy theory. The key move was to replace raw textual mentions by objects defined only up to weak equivalence. To do this, we introduced a syntax category of causal mentions, a causal semantics category of normalized claims, and a causal extraction functor

$$E : \mathcal{S}_{\text{causal}} \rightarrow \mathcal{C}_{\text{causal}}.$$

Weak equivalences were defined as those morphisms that become isomorphisms after applying E , and the resulting homotopy category

$$\text{Ho}(\mathcal{S}_{\text{causal}}) = \mathcal{S}_{\text{causal}}[W_E^{-1}]$$

captures causal mentions modulo paraphrase-preserving normalization.

We then introduced lifting problems, simplicial constructions, horns, Kan complexes, and model categories as the abstract machinery needed to reason about such equivalence classes. The conceptual lesson is that causality extracted from language should not be modeled directly at the level of unstable surface forms. Instead, it should be studied through localized, liftable, and higher-order coherent structures. In this sense, model categories provide a concise vocabulary for both paraphrase invariance in language and semantic stability in causal inference. Establishing a domain-specific model structure remains a separate theorem.

Suggested Reading

Quillen's original framework [Quillen, 1967] and Hovey [1999] provide the standard foundations of model categories. For simplicial sets and horn-filling, see May [1992]. Readers interested in causal semantics should compare the string-diagrammatic account of Jacobs et al. [2021].

Predictive State Representations in a Topos

Previously, we introduced *Topos Causal Models* (TCM) as a framework for representing causal structure across multiple contexts. In that setting, a system is not described by a single global model, but by a family of local causal descriptions together with compatibility conditions that determine when these local views can be glued into a coherent whole.

In the present chapter we extend this perspective from causal structure to *predictive state*. Rather than attempting to infer a single latent state for a partially observed system, we represent state by a family of *context-dependent predictive summaries*. Each local context supports its own action–observation tests, and state is given by the probabilities of these tests. The central question then becomes whether these local predictive states are compatible and can be glued into a global predictive description.

This shift is particularly natural in the corporate setting studied in the previous chapters. A 10-K filing does not provide a complete description of a firm’s internal state; instead, it offers multiple partial views arising from different operational contexts such as supply chain, pricing, distribution, innovation, and capital allocation. Subsequent earnings reports and market signals provide further observations, but again only locally and with noise. It is therefore more appropriate to represent the firm through a family of local predictive states rather than a single hidden variable.

To construct these predictive states, we build on the compositional workflow framework developed in the BASKET and ROCKET chapters. BASKET extracts normalized operational actions from text, while ROCKET selects workflows aligned with realized outcomes. These workflows serve as the *actions* in a predictive state representation, while financial and textual signals provide the corresponding observations.

We thus obtain a unified picture:

- **Topos Causal Models** provide the space of contexts and gluing structure,
- **BASKET** provides compositional actions extracted from text,
- **ROCKET** provides outcome-aligned workflow selection,
- **Predictive State Representations** assign to each context a predictive state over future action–observation tests.

In this formulation, predictive state becomes a *section of a sheaf over contexts*. Compatibility of these sections encodes coherence of the system, while failures of gluing provide a principled notion of inconsistency between local views. This yields a compositional, context-sensitive, and predictive semantics for agentic systems that directly extends the categorical framework developed in earlier chapters.

Predictive State Representations

Traditional latent-state models assume that a hidden state x_t generates observations. In contrast, predictive state representations (PSRs) define state directly in terms of predictions of future observable tests.

Action–Observation Tests

Let $\mathcal{A}$ be a set of actions and $\mathcal{O}$ a set of observations. A *test* is a finite sequence

$$\tau = (a_1, o_1, \dots, a_k, o_k),$$

where $a_i \in \mathcal{A}$ and $o_i \in \mathcal{O}$.

In the corporate setting:

- actions are normalized workflow steps (e.g., `optimize`, `expand`, `market`),
- observations are downstream financial or textual signals (e.g., revenue growth, margin improvement, guidance changes).

Predictive State

Given a history h_t , the predictive state is defined as

$$q_t = (\Pr(\tau_1 \mid h_t), \dots, \Pr(\tau_m \mid h_t)),$$

for a chosen set of tests $\{\tau_1, \dots, \tau_m\}$.

Thus state is represented not by a hidden variable, but by the system's predictions about future action–observation sequences.

Workflows as Intervention Sequences

In the present framework, workflows extracted by `BASKET` are interpreted as structured interventions.

Judo Calculus Interpretation

Each normalized action $a \in \mathcal{A}$ is interpreted as an intervention operator

$$J_a : X \rightarrow X,$$

and a workflow

$$a_1 \rightarrow a_2 \rightarrow \cdots \rightarrow a_k$$

corresponds to the composed intervention

$$J_{a_k} \circ \cdots \circ J_{a_1}.$$

Thus corporate plans become compositional intervention programs, and PSR tests become predictions about the outcomes of these programs.

Predictive State Representations as Sheaf Sections

In the topos-theoretic framework of the previous chapter, a system is described not by a single global object but by a family of local objects indexed by contexts, together with restriction maps and gluing conditions. We now show that predictive state representations can be naturally formulated in this same language.

Action–Observation Tests

Let $\mathcal{A}$ denote a set of actions and $\mathcal{O}$ a set of observations. A *test* is a finite sequence

$$\tau = (a_1, o_1, \dots, a_k, o_k),$$

with $a_i \in \mathcal{A}$ and $o_i \in \mathcal{O}$.

In the present setting, actions are drawn from normalized workflows extracted by BASKET (e.g., optimize, expand, market), while observations correspond to downstream financial or textual signals (e.g., revenue growth, margin improvement, or guidance revisions).

Local Predictive State Spaces

Let $\mathcal{C}$ be a category of contexts (for example, operations, market, or finance). To each context $U \in \mathcal{C}$ we associate a vector space

$$\mathcal{F}(U),$$

whose elements are predictive states over tests observable within U .

Concretely, a section $s_U \in \mathcal{F}(U)$ assigns to each test τ supported in U a probability

$$s_U(\tau) = \Pr(\tau \mid h_t),$$

where h_t is the observed history up to time t .

Thus predictive state is not a single global vector but a family of local sections

$$\{s_U\}_{U \in \mathcal{C}}.$$

Restriction Maps

If $V \subseteq U$ is a refinement of contexts, we define restriction maps

$$\rho_{U,V} : \mathcal{F}(U) \rightarrow \mathcal{F}(V),$$

which project predictive states by restricting attention to tests supported in V .

Intuitively, $\rho_{U,V}$ forgets predictions that cannot be expressed within the smaller context.

Sheaf Condition

The assignment $U \mapsto \mathcal{F}(U)$ defines a presheaf of predictive states. It becomes a sheaf if the following gluing condition holds:

Given a family of local sections $\{s_U\}$ that agree on overlaps,

$$\rho_{U,U \cap V}(s_U) = \rho_{V,U \cap V}(s_V),$$

there exists a unique global section s such that

$$\rho_U(s) = s_U \quad \text{for all } U.$$

In this setting, a global predictive state corresponds to a coherent assignment of test probabilities across all contexts.

Interpretation

This formulation yields a different notion of state from classical models:

- State is *predictive*: it encodes probabilities of future action–observation tests.
- State is *local*: it is defined separately in each context.
- State is *compositional*: it is assembled from local sections via gluing.

Importantly, global state need not always exist. When local predictive states fail to agree on overlaps, no global section can be constructed.

Failure of Gluing as a Diagnostic

Failure of the sheaf condition has a natural interpretation in the corporate domain. If local predictive states derived from different contexts disagree, this may indicate:

- inconsistency between operational improvements and market outcomes,
- conflicting signals across business units,
- incomplete or misleading disclosures,
- structural instability in the firm’s execution pipeline.

Thus predictive inconsistency becomes an observable phenomenon rather than a modeling error.

Relation to Workflow Extraction

The predictive state construction is grounded in the workflow framework developed earlier:

- BASKET provides the action alphabet and compositional structure,
- ROCKET selects workflows aligned with observed outcomes,
- predictive state assigns probabilities to tests built from these actions.

In this way, predictive state representations arise naturally as sections of a sheaf over the space of operational contexts.

Local Predictive States

Corporate disclosures arise from multiple local contexts: operations, pricing, distribution, innovation, and so on. Each context supports its own action vocabulary and observation signals.

We therefore associate to each context U a predictive state space

$$\mathcal{F}(U),$$

whose elements are predictions of tests visible within U .

For example:

- in an operations context, tests may involve margin improvements,
- in a market context, tests may involve revenue growth,
- in a finance context, tests may involve leverage or return metrics.

Thus instead of a single global state, we obtain a family of local predictive states $\{q_t^U\}$.

Sheaf of Predictive States

These local predictive states naturally form a presheaf.

Restriction Maps

For contexts $V \subseteq U$, we define restriction maps

$$\rho_{U,V} : \mathcal{F}(U) \rightarrow \mathcal{F}(V),$$

which project predictive states onto smaller contexts.

Gluing

A family of local states $\{s_U\}$ is *compatible* if it agrees on overlaps:

$$\rho_{U, U \cap V}(s_U) = \rho_{V, U \cap V}(s_V).$$

When such compatibility holds, the states can be glued into a global predictive state.

Failure of Gluing

If compatibility fails, no global state exists. This is not an error but a signal:

- inconsistencies in corporate strategy,
- mismatches between operational and market signals,
- incomplete or conflicting disclosures.

Thus failure of gluing provides a diagnostic for structural incoherence.

A Judo Calculus for Predictive Tests

The workflow actions extracted by BASKET can be interpreted as composable interventions, and predictive state representations assign probabilities to the outcomes of such interventions. This suggests a natural extension of judo calculus to the predictive setting, in which rules govern how action compositions transform predictive states.

Rather than introducing a full formal system, we describe a small set of principles that capture the essential structure of predictive intervention semantics.

ACTIONS AS INTERVENTIONS. Each normalized action $a \in \mathcal{A}$ is interpreted as an intervention operator

$$J_a : X \rightarrow X,$$

and a workflow

$$a_1 \rightarrow a_2 \rightarrow \cdots \rightarrow a_k$$

corresponds to the composed intervention

$$J_{a_k} \circ \cdots \circ J_{a_1}.$$

Thus predictive tests describe the outcomes of compositional intervention programs.

(J1) **IDENTITY.** The empty intervention leaves predictive state unchanged:

$$\Pr(\tau \mid h_t) = \Pr(\tau \mid \text{id}, h_t).$$

(J2) **SEQUENTIAL COMPOSITION.** Sequential interventions compose associatively. For actions a and b ,

$$\Pr(\tau \mid a; b, h_t) = \Pr(\tau \mid b, a, h_t),$$

so that predictive tests depend only on the composed intervention.

(J3) **PREDICTIVE UPDATE.** After observing an action–observation pair (a, o) , predictive state updates by conditioning:

$$\Pr(\tau \mid h_t, a, o) = \frac{\Pr((a, o) \circ \tau \mid h_t)}{\Pr(a, o \mid h_t)}.$$

This is the predictive analogue of Bayesian or Radon–Nikodym updating.

(J4) **MONOIDAL COMPOSITION.** Independent or weakly coupled actions combine via the monoidal product. For actions a and b ,

$$\Pr(\tau \mid a \otimes b, h_t) = \Phi(\Pr(\tau \mid a, h_t), \Pr(\tau \mid b, h_t)),$$

where Φ is a combination operator encoding independence or interaction structure. This reflects the symmetric monoidal structure of workflows.

(J5) **GLUING COMPATIBILITY.** For contexts U and V , predictive states must agree on overlaps:

$$\rho_{U, U \cap V}(s_U) = \rho_{V, U \cap V}(s_V),$$

which holds precisely when the corresponding intervention predictions agree on shared tests. This condition determines whether local predictive states can be glued into a global section.

INTERPRETATION. These rules define a compositional calculus in which:

- workflows act as structured interventions,
- predictive states assign probabilities to their outcomes,
- composition governs how interventions combine,
- and sheaf compatibility governs global coherence.

In this way, predictive state representations become functorial assignments from compositional intervention diagrams to probabilities, integrating judo calculus, symmetric monoidal structure, and sheaf-theoretic gluing into a single framework.

A Toy Example

Consider two contexts:

- U : operations (optimize),
- V : market (expand, market).

Local predictive states:

$$q_U = \Pr(\text{margin-up} \mid \text{optimize}),$$

$$q_V = \Pr(\text{revenue-up} \mid \text{expand}).$$

On the overlap, we consider the composite test:

$$\text{optimize} \rightarrow \text{expand}.$$

If the predictions agree, the states glue. If not, the discrepancy indicates a mismatch between operational improvements and market outcomes.

A Worked Nike Example

We illustrate the predictive-state and sheaf-theoretic framework using Nike, whose extracted workflows provide a clear example of compositional operational structure. A representative displayed path in Nike's post-ROCKET workflows is

$$\text{digitize} \rightarrow \text{optimize} \rightarrow \text{expand} \rightarrow \text{market} \rightarrow \text{realize\_revenue}.$$

As emphasized earlier, this is not the plan itself but a linear extension of a richer partially ordered workflow.

Actions as Interventions

Each action is interpreted as a judo intervention operator. The workflow above corresponds to the composed intervention

$$J_{\text{realize\_revenue}} \circ J_{\text{market}} \circ J_{\text{expand}} \circ J_{\text{optimize}} \circ J_{\text{digitize}}.$$

By (J2), predictive tests depend only on this composed intervention, not on the particular parenthesization of the sequence.

Predictive Tests

We define a small set of predictive tests for Nike:

$$\tau_1 = (\text{optimize}, \text{margin\_up}),$$

$$\tau_2 = (\text{expand}, \text{revenue\_up}),$$

$$\tau_3 = (\text{market}, \text{channel\_up}).$$

together with a composite test

$$\tau_4 = (\text{digitize}, \text{channel\_up}), (\text{optimize}, \text{margin\_up}), (\text{expand}, \text{revenue\_up}).$$

The predictive state assigns probabilities

$$q_i^{\text{Nike}}(\tau_i) = \Pr(\tau_i \mid h_t),$$

where h_t denotes Nike's history up to time t .

Predictive Update

Suppose a filing and subsequent earnings report produce the observation pair

$$(\text{optimize}, \text{margin\_up}).$$

Then, by (J3), predictive state updates as

$$\begin{aligned} & \Pr(\tau \mid h_t, \text{optimize}, \text{margin\_up}) \\ &= \frac{\Pr((\text{optimize}, \text{margin\_up}) \circ \tau \mid h_t)}{\Pr(\text{optimize}, \text{margin\_up} \mid h_t)}. \end{aligned}$$

Thus observed execution outcomes directly refine predictions of future tests.

Local Contexts

We now decompose the system into two local contexts:

$$\begin{aligned} U_{\text{ops}} &= \{\text{digitize}, \text{optimize}, \text{distribute}\}, \\ U_{\text{market}} &= \{\text{expand}, \text{market}, \text{realize\_revenue}\}. \end{aligned}$$

In the operations context, predictive tests include

$$(\text{digitize}, \text{channel\_up}), \quad (\text{optimize}, \text{margin\_up}),$$

yielding a local predictive state $s_{U_{\text{ops}}}$.

In the market context, tests include

$$\begin{aligned} & (\text{expand}, \text{revenue\_up}), \\ & (\text{market}, \text{channel\_up}), \\ & (\text{realize\_revenue}, \text{guidance\_up}). \end{aligned}$$

yielding a local predictive state $s_{U_{\text{market}}}$.

Monoidal Composition

The interaction between contexts is captured by monoidal composition. For example, the partially ordered workflow

$$(\text{digitize} \otimes \text{optimize}) \rightarrow \text{expand}$$

combines independent operational improvements before expansion. By (J4), the predictive effect of this joint intervention is obtained by combining the individual predictive contributions of `digitize` and `optimize`.

Gluing and Consistency

The overlap between contexts corresponds to the transition

$$\text{optimize} \rightarrow \text{expand}.$$

The sheaf condition (J5) requires that predictions of outcomes along this transition agree between contexts. Concretely, both $s_{U_{\text{ops}}}$ and $s_{U_{\text{market}}}$ induce predictions for

$$\Pr(\text{revenue\_up} \mid \text{optimize} \rightarrow \text{expand}).$$

If these agree, the local predictive states glue into a global section. If they disagree, no global predictive state exists.

Interpretation

This distinction has a natural economic interpretation. If operational improvements (captured by `optimize`) predict strong downstream expansion effects, and market observations confirm this, the predictive states are compatible. If not, the failure of gluing indicates a mismatch between internal execution and external commercial performance.

Role of ROCKET

Finally, this example clarifies the role of ROCKET. The transformation

$$\begin{array}{c} \text{digitize} \rightarrow \text{realize\_revenue} \longrightarrow \text{digitize} \rightarrow \text{optimize} \rightarrow \text{expand} \\ \hspace{15em} \rightarrow \text{market} \rightarrow \text{realize\_revenue}. \end{array}$$

introduces intermediate interventions that make predictive tests more informative. In particular, the inserted actions expose the transition

$$\text{optimize} \rightarrow \text{expand},$$

which is precisely the overlap on which gluing is evaluated.

Thus ROCKET not only improves workflow plausibility but also refines the predictive structure of the system, enabling more coherent sheaf-based state estimation.

Summary

Nike illustrates the full framework:

- workflows define compositional interventions (J1–J2),
- predictive states update under observations (J3),
- independent actions combine via monoidal structure (J4),
- and coherence is determined by gluing conditions (J5).

In this way, predictive state representations become a compositional, context-dependent semantics for corporate strategy.

A Failure-to-Glue Scenario

The sheaf formulation allows us to distinguish between coherent and incoherent predictive structure. We now consider a hypothetical but realistic failure case for Nike, illustrating how incompatibility of local predictive states manifests as a failure of gluing.

CONFLICTING LOCAL PREDICTIONS. Suppose that in the operations context U_{ops} , recent filings and execution signals suggest that

$$\Pr(\text{margin\_up} \mid \text{optimize}) \text{ is high,}$$

and that operational improvements are expected to enable expansion:

$$\Pr(\text{expand} \mid \text{optimize}) \text{ is high.}$$

Thus the operations-side predictive state supports the transition

$$\text{optimize} \rightarrow \text{expand}.$$

However, suppose that in the market context U_{market} , observed outcomes indicate weak demand or channel performance, leading to

$$\Pr(\text{revenue\_up} \mid \text{expand}) \text{ is low.}$$

OVERLAP INCONSISTENCY. The overlap between the two contexts involves the composite test

$$\text{optimize} \rightarrow \text{expand} \rightarrow \text{revenue\_up}.$$

From the operations perspective, this transition should increase the likelihood of downstream success. From the market perspective, it does not.

Formally, the induced predictions satisfy

$$\rho_{U_{\text{ops}}, U_{\text{ops}} \cap U_{\text{market}}}(s_{U_{\text{ops}}}) \neq \rho_{U_{\text{market}}, U_{\text{ops}} \cap U_{\text{market}}}(s_{U_{\text{market}}}),$$

so the compatibility condition (J5) fails.

INTERPRETATION. This failure of gluing has a clear economic meaning. The firm's internal operational improvements are not translating into external commercial outcomes. Possible interpretations include:

- supply-chain or cost efficiencies are not visible to customers,
- expansion efforts are misaligned with market demand,
- pricing or channel strategy is ineffective,
- macroeconomic conditions suppress revenue despite internal improvements.

RELATION TO WORKFLOW STRUCTURE. Notably, the workflow itself may appear well-formed:

`digitize → optimize → expand → market → realize_revenue.`

However, the predictive semantics of this workflow are inconsistent across contexts. The failure is therefore not structural but semantic: the composed intervention does not produce the expected outcomes.

DIAGNOSTIC ROLE OF ROCKET. In such cases, ROCKET may respond by modifying the workflow. For example, it may insert additional steps such as `market` or `distribute`, yielding

`optimize → expand → market → realize_revenue,`

in an attempt to restore predictive coherence. Alternatively, it may reduce confidence in the workflow altogether.

SUMMARY. Failure of gluing provides a principled notion of predictive inconsistency. Rather than treating such discrepancies as noise, the sheaf formulation interprets them as meaningful signals about the relationship between internal execution and external outcomes. This turns predictive state estimation into a diagnostic tool for evaluating the coherence of corporate strategy.

Subobject Classifiers and Intuitionistic Predictive Semantics

The sheaf-based formulation of predictive state admits a natural interpretation in terms of the internal logic of a topos. In this setting, predictive tests are not simply assigned numerical probabilities, but are evaluated as generalized truth values in a subobject classifier.

SUBOBJECT CLASSIFIER. Let Ω denote the subobject classifier of the topos of predictive states. For each context U , a predictive state assigns to each test τ a truth value

$$\chi_U(\tau) \in \Omega(U),$$

interpreted as the extent to which the test τ holds in context U .

Thus, instead of a probability

$$\Pr(\tau \mid h_t),$$

we obtain a section

$$\chi : \tau \mapsto \chi_U(\tau),$$

which can be viewed as a generalized truth assignment.

INTUITIONISTIC SEMANTICS. Unlike classical truth values, elements of $\Omega(U)$ form a Heyting algebra. Logical operations on predictive tests therefore follow intuitionistic rules:

- conjunction $\tau_1 \wedge \tau_2$ corresponds to joint satisfaction,
- disjunction $\tau_1 \vee \tau_2$ corresponds to alternative predictive paths,
- implication $\tau_1 \Rightarrow \tau_2$ expresses predictive entailment.

In particular, implication provides a natural way to express statements such as:

$$\text{optimize} \Rightarrow \text{margin\_up},$$

meaning that operational improvement predicts margin improvement.

CONTEXT DEPENDENCE. Truth values depend on context. A test may be highly supported in one context and weakly supported in another:

$$\chi_{U_{\text{ops}}}(\tau) \neq \chi_{U_{\text{market}}}(\tau).$$

This reflects the fact that predictive validity is inherently local.

GLUING AS LOGICAL CONSISTENCY. The sheaf condition now acquires a logical interpretation. A family of local truth assignments $\{\chi_U\}$ is compatible if the truth values agree on overlaps:

$$\rho_{U, U \cap V}(\chi_U) = \rho_{V, U \cap V}(\chi_V).$$

When this holds, the assignments glue to a global truth valuation.

Failure of gluing corresponds to logical inconsistency: a test may be “true” in one context and “false” in another. This is precisely the situation described in the previous section, where operational and market predictions disagree.

RELATION TO JUDO CALCULUS. The judo rules (J1–J5) now admit an internal logical interpretation. Each intervention induces a transformation of truth values:

$$\chi_U(\tau) \mapsto \chi_U(a \cdot \tau),$$

and composition corresponds to the functorial action of intervention diagrams on Ω -valued predicates.

INTERPRETATION. This perspective replaces probabilistic prediction with intuitionistic truth. Predictive state becomes a logical object: a section of the subobject classifier assigning truth values to tests. The coherence of the system is then determined not only numerically but logically, through the compatibility of these truth assignments across contexts.

RECOVERABILITY CONJECTURE FOR TCM-PSRS. Let $\mathcal{C}$ be a context category, and suppose that each context $U \in \mathcal{C}$ carries a hidden local state space X_U together with a finite-dimensional predictive representation $\mathcal{F}(U)$ generated by a family of tests $\mathcal{T}_U$. Assume:

1. (*local separation*) for any distinct reachable states $x, x' \in X_U$, there exists a test $\tau \in \mathcal{T}_U$ such that

$$\Pr(\tau \mid x) \neq \Pr(\tau \mid x');$$

2. (*finite local rank*) the predictive system on each U has finite linear dimension d_U , so that $\mathcal{T}_U$ contains a core basis of size d_U ;
3. (*descent compatibility*) the predictive sections and hidden local states agree under restriction on overlaps.

Then the hidden local state on each context U is recoverable up to predictive equivalence from the local predictive section $s_U \in \mathcal{F}(U)$. Moreover, if the family $\{s_U\}$ satisfies the sheaf gluing condition, then the local hidden states glue to a global hidden state object whenever such a global object exists.

Descent Defects and a Cohomological Caveat

The sheaf-theoretic formulation naturally leads one to ask whether failure of reconstruction defines a cohomological obstruction. For the elementary vector-valued mismatch below, the answer requires care.

MISMATCH ON OVERLAPS. Given local predictive states $s_U \in \mathcal{F}(U)$ and $s_V \in \mathcal{F}(V)$, their restrictions to the overlap $U \cap V$ need not agree. Define the mismatch

$$\delta_{UV} = \rho_{U, U \cap V}(s_U) - \rho_{V, U \cap V}(s_V).$$

MISMATCH COCHAIN. For a cover $\{U_i\}$, the collection $\{\delta_{ij}\}$ forms a Čech 1-cochain. Because it is the coboundary of the 0-cochain of local sections, the alternating sum on triple overlaps is zero:

$$\delta_{ij} + \delta_{jk} + \delta_{ki} = 0,$$

Thus it is a cocycle, but also an exact one. Its cohomology class therefore cannot by itself be a nonzero obstruction.

OBSTRUCTION TO GLUING. The given local predictive states glue precisely when every mismatch δ_{ij} is zero and the presheaf satisfies the sheaf condition for the chosen cover. A nonzero mismatch is therefore a direct descent defect. A genuine cohomological obstruction requires a different extension problem (for example, torsor-valued or transition data) rather than the raw coboundary of a family of vector-valued local sections.

INTERPRETATION. Thus failure of predictive-state reconstruction can be diagnosed by an overlap mismatch. Even when each local context admits a finite predictive representation, the global hidden state may fail to exist due to incompatibility across contexts.

RELATION TO EARLIER WORK. This resembles the obstruction-based diagnostics developed in the GT+DB framework, but the present mismatch is only a descent diagnostic. A genuine cohomology class would have to be constructed from non-exact transition or extension data.

Theorem 21 (Local predictive coordinates and exact descent). *Let $\mathcal{C}$ be a context category, let $\{U_i\}_{i \in I}$ be a finite cover of relevant operational contexts, and let*

$$\mathcal{F} : \mathcal{C}^{op} \rightarrow \mathbf{Vect}$$

be the presheaf of local predictive-state spaces, where each $\mathcal{F}(U_i)$ is generated by a finite family of predictive tests $\mathcal{T}_{U_i}$.

Assume the following conditions.

1. **Local separation.** *For each context U_i , the chosen test family $\mathcal{T}_{U_i}$ separates reachable hidden local states: if $x, x' \in X_{U_i}$ are distinct reachable local states, then there exists a test $\tau \in \mathcal{T}_{U_i}$ such that*

$$\Pr(\tau \mid x) \neq \Pr(\tau \mid x').$$

2. **Finite local predictive rank.** *For each U_i , the local predictive Hankel matrix*

$$H_{U_i}(h, \tau) = \Pr(\tau \mid h)$$

has finite rank $d_i < \infty$, so that $\mathcal{F}(U_i)$ admits a basis of d_i core tests.

3. **Compatible restrictions.** *For each overlap $U_i \cap U_j$, the restriction maps*

$$\rho_{U_i, U_i \cap U_j} : \mathcal{F}(U_i) \rightarrow \mathcal{F}(U_i \cap U_j)$$

are well-defined, and the local predictive states

$$s_i \in \mathcal{F}(U_i)$$

induce overlap mismatches

$$\delta_{ij} = \rho_{U_i, U_i \cap U_j}(s_i) - \rho_{U_j, U_i \cap U_j}(s_j)$$

that satisfy the cocycle condition on triple overlaps:

$$\delta_{ij} + \delta_{jk} + \delta_{ki} = 0.$$

Then:

- (a) *each hidden local state on U_i is recoverable up to predictive equivalence from the local predictive section $s_i \in \mathcal{F}(U_i)$;*
- (b) *if $\mathcal{F}$ is a sheaf for the chosen cover, the family $\{s_i\}$ glues to a unique global predictive state if and only if $\delta_{ij} = 0$ on every pairwise overlap;*
- (c) *whenever $\delta = 0$ and the hidden local states satisfy effective descent, the recovered local hidden states glue to a global hidden state object.*

Proof sketch. Part (a) is the local predictive-state reconstruction statement. By local separation, distinct reachable hidden local states induce distinct predictive functionals on the chosen test family. By finite local predictive rank, each local predictive system admits a finite core basis, so the local hidden state is recoverable up to predictive equivalence from the corresponding finite predictive coordinates.

For part (b), define the overlap mismatch cochain

$$\delta_{ij} = \rho_{U_i, U_i \cap U_j}(s_i) - \rho_{U_j, U_i \cap U_j}(s_j).$$

The local sections form compatible descent data exactly when this cochain is zero. The existence and uniqueness clauses of the sheaf axiom then give the global section. Conversely, restrictions of a global section agree on every overlap, so all mismatches vanish. Notice that the weaker condition $[\delta] = 0$ would not be enough to glue the original sections: here $\delta = ds$ is already a coboundary.

Part (c) follows by descent. Once the predictive sections glue and local hidden states are recoverable from them, any descent-compatible family of local hidden states assembles to a global hidden state object whenever such descent data are effective. $\square$

Theorem 22 (Classical PSR as a special case). *If the context cover consists of a single context U , there are no nontrivial overlap conditions, and Theorem 21 reduces to the classical finite-rank PSR recoverability statement: a finite-dimensional predictive representation recovers hidden state up to predictive equivalence.*

INTERPRETATION. Theorem 21 isolates one new ingredient beyond the classical local theory: descent. Local predictive reconstruction is controlled by local rank and test separation, as in ordinary PSRs, while global reconstruction requires exact agreement on overlaps and effectiveness of the hidden-state descent data. Thus hidden state is recoverable when local predictive models are both sufficient and globally compatible.

Definition 86 (Predictive mismatch diagnostic). *The overlap family $\delta = \{\delta_{ij}\}$ associated with local predictive states is called the predictive mismatch diagnostic. For a sheaf, the given local states glue if and only if this cochain is identically zero. Its norm can be used as an approximate compatibility statistic, but it is not a nontrivial cohomology class.*

Relation to BASKET and ROCKET

The full pipeline now becomes:

- BASKET: extracts compositional workflows from text,
- ROCKET: selects workflows aligned with outcomes,
- PSR: constructs predictive states from action–observation tests,
- Sheaf layer: glues local predictive states into global structure.

Thus BASKET provides actions, ROCKET provides evaluation, and PSR provides state estimation.

Connection to Topos Causal Models

The sheaf of predictive states provides a bridge to topos-theoretic causal models. Each context can be viewed as a local logical universe in which predictions are evaluated. The global structure, when it exists, is obtained by gluing these local logics.

This connects predictive modeling, causal reasoning, and compositional structure within a single categorical framework.

Experimental Results

We now summarize the first empirical comparison of the predictive-state layer within the BASKET–ROCKET 10-K workflow pipeline. The goal of these experiments is not yet to establish a final benchmark, but rather to evaluate whether the sheaf-theoretic PSR construction produces meaningful local tests, whether longer compositional tests improve predictive quality, and whether section-aware workflow extraction and KET reranking alter the resulting predictive structure in a systematic way.

What the Implementation Measures

The implementation uses four hand-specified local contexts: *operations*, *market*, *finance*, and *innovation*. For each company-year, the extracted workflow is projected into each context by retaining only context-supported actions and collapsing adjacent repetitions. The PSR catalog is then constructed from contiguous action subsequences up to a chosen maximum length, paired with a single downstream financial observation such as `margin_up`, `revenue_up`, `free_cash_flow_up`, `roa_up`, or `leverage_down`.

Thus each catalog entry is a compressed test of the form

$$a_1 \rightarrow a_2 \rightarrow \cdots \rightarrow a_k \Rightarrow o,$$

Variant	Tests	Avg tests/state	Glueable	Brier	AUC	Gap
Baseline	324	37.68	585/588	0.2030	0.6393	0.0605
Max length 3	336	44.60	588/588	0.2031	0.6412	0.0618
Section-aware	324	39.94	581/588	0.2017	0.6459	0.0629

which should be interpreted as a marginalization over full action–observation tests

$$[(a_1, o_1), (a_2, o_2), \dots, (a_k, o)],$$

for admissible intermediate observations $o_1, \dots, o_{k-1}$. In this sense, the implementation realizes a tractable approximation to the full PSR test space.

Predictive probabilities are estimated by blending company-level, sector-level, and panel-level frequencies with leave-one-out smoothing. In sheaf-theoretic terms, each company-year yields a family of local predictive sections $\{s_U\}$.

Restriction maps are implemented concretely by dropping tests unsupported in the target context. Pairwise gluing is evaluated by comparing predictive values on overlaps; a pair is declared *glueable* when the maximum absolute mismatch is at most 0.18. The stored “global section” is therefore an empirical surrogate obtained by averaging compatible local sections, rather than a proof of exact sheaf-theoretic descent.

PSR Variant Comparison

The table above compares three PSR variants: a baseline catalog with maximum test length 2, a longer-test variant with maximum test length 3, and a section-aware variant in which extraction retains section information before PSR construction.

Several points are worth emphasizing.

Table 51: Summary of PSR variants on the 10-K panel. “Glueable” reports the number of company-years whose local predictive sections are compatible under the overlap-mismatch tolerance.

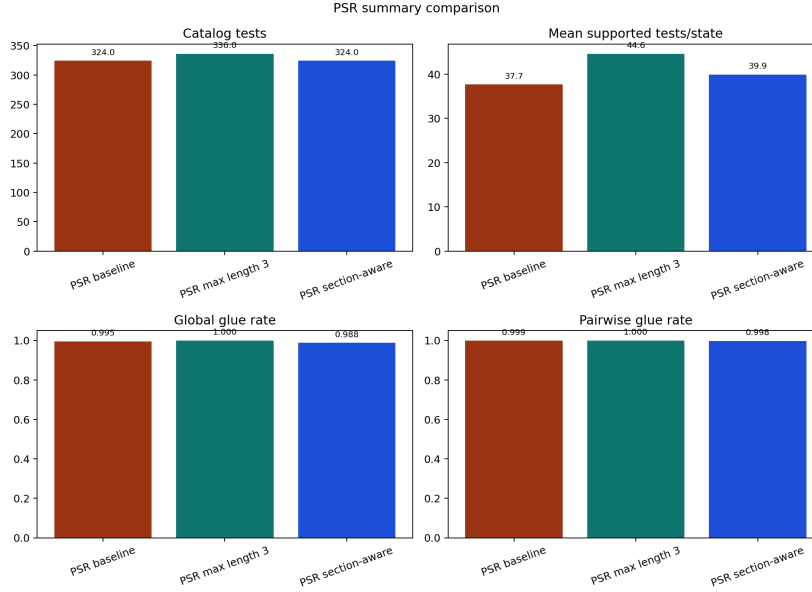


Figure 57: Summary comparison of PSR variants. The top panels measure local expressive power (test catalog size and mean supported tests per state), while the bottom panels measure global compatibility (glue rates). The max-length-3 variant achieves the highest local coverage and perfect gluing, while the section-aware variant trades a small reduction in glueability for improved predictive discrimination.

Experiment: predictive-state variant comparison

First, increasing test length increases *catalog coverage*. The catalog grows modestly, the number of supported tests per state increases, and the empirical gluing rate reaches 588/588. In categorical terms, longer tests provide a richer candidate test family. Catalog size is not the rank of a predictive Hankel matrix, and the experiment does not estimate that rank. Predictive differences are small: the bootstrap confidence intervals for Brier score and AUC overlap substantially across all three variants. The run therefore does not establish that the longer family improves prediction.

Second, the section-aware PSR has the best point estimates for Brier score, AUC, and probability gap, even though its thresholded glueability drops slightly. The bootstrap intervals overlap, so this is a descriptive ordering rather than evidence of a statistically resolved advantage. It is compatible with a local-resolution/global-compatibility trade-off, but does not by itself identify that mechanism.

The below figure provides a compact view of the tradeoff between local predictive richness and global compatibility. Increasing test length expands the predictive basis, raising both the number of catalog tests and the mean number of supported tests per state. This increases the number of available test coordinates; it should not be identified with the local predictive rank d_U without computing the rank of the associated Hankel operator.

At the same time, the glue rates measure agreement only at the fixed toler-

Variant	Changed rate	Mean score gain	Mean company score	Pos. comps.	Neg. comps.
Baseline	19.82%	0.00378	0.239	14	10
Section-aware	17.95%	0.00306	0.177	16	8
Section-aware + KET	23.07%	0.00729	-0.066	8	16

Table 52: ROCKET reranking comparison.

ance 0.18. The max-length-3 variant passes all 588 thresholded checks; this is not exact descent and does not imply that every overlap mismatch vanishes. The section-aware variant slightly reduces the thresholded glue rate. This is consistent with finer context distinctions exposing more mismatches, but the experiment does not isolate that explanation.

Thus the figure illustrates two measured ingredients of predictive-state reconstruction: local sufficiency (captured by test richness) and global compatibility (captured by thresholded gluing). The point estimates show that these statistics can move differently; a designed intervention or uncertainty-resolved comparison would be needed to establish a general trade-off.

The experiment does not directly estimate the theoretical error decomposition. It records thresholded overlap mismatches and predictive scores under one chosen context system and tolerance. A tolerance sweep and an explicit association between mismatch magnitude and held-out loss would be needed to test whether the diagnostic explains predictive error.

ROCKET Variant Comparison

The table above summarizes the corresponding ROCKET comparison.

The section-aware + KET variant changes the most workflows and has the largest gain under the constructed reranking score, but its mean company ranking score is negative. These are different objectives. The bootstrap interval for each mean company score includes zero, so the table does not establish an improvement or degradation in realized company outcomes.

The below figure provides a panel-level view of the effect of workflow reranking. Unlike aggregate statistics, this visualization reveals substantial heterogeneity in the constructed company scores. Positive and negative values should not be called realized benefits or harms without an independent outcome evaluation.

This pattern is compatible with company-specific predictive mismatch, but the heatmap is not itself a mismatch measurement and does not identify its cause. The company drilldowns are therefore useful hypotheses for a later mismatch analysis, not estimates of the obstruction term in the error bound.

Representative Predictive Tests

The table below shows representative tests.

The baseline is dominated by short, high-support tests, while longer tests introduce richer compositional structure. The section-aware variant reveals

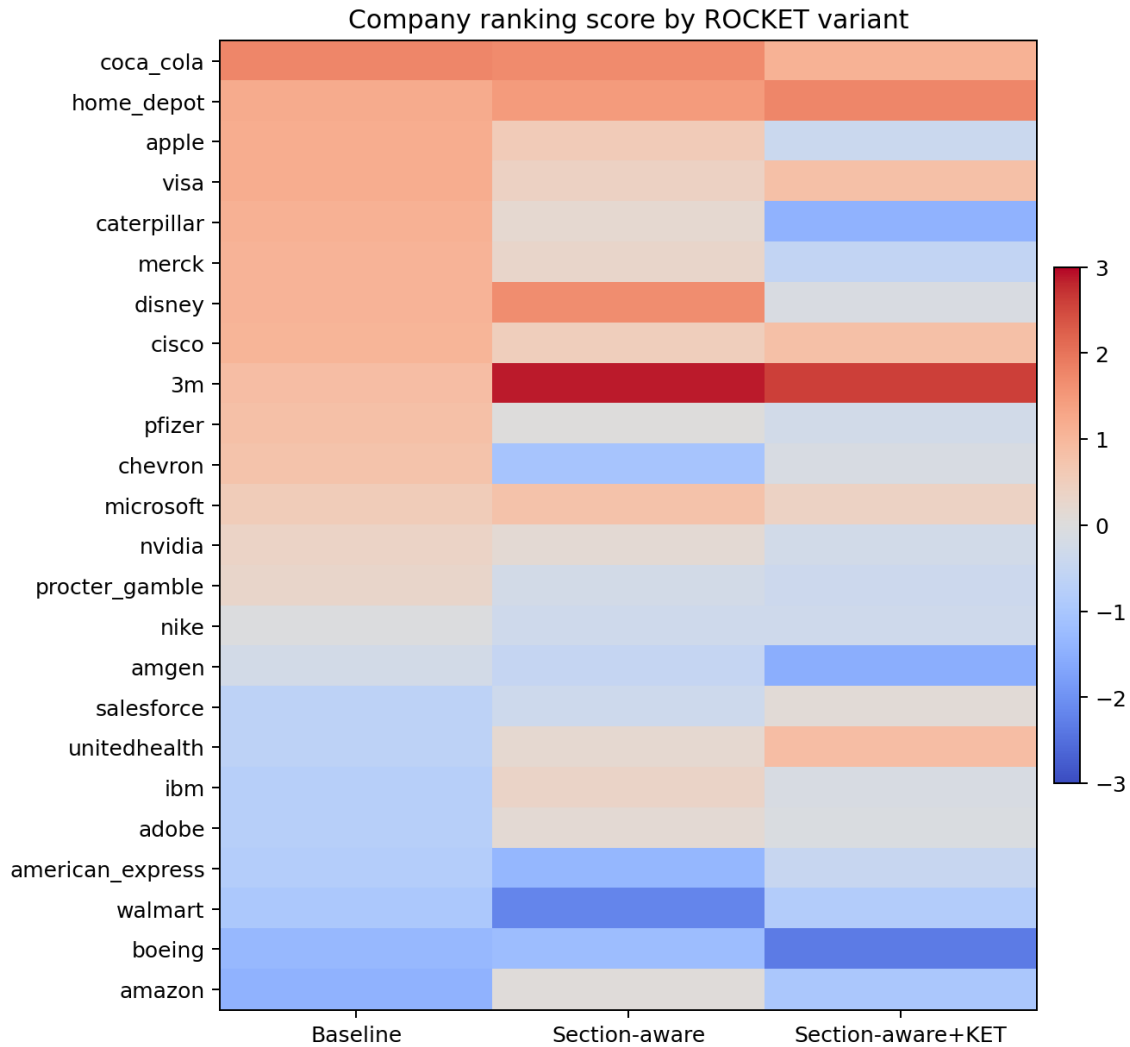


Figure 58: Company-level ranking scores across ROCKET variants. Each row corresponds to a company, and each column to a reranking variant. The strong variation across companies shows that the constructed ranking score is not uniform across the panel.

context-specific patterns, but also exposes non-predictive tests, illustrating the need for both extraction and selection.

Gluing and Predictive Obstruction

The glueability statistic in the above table provides a thresholded compatibility diagnostic inspired by the sheaf condition. When a company-year is “glueable,” the local predictive states across contexts are mutually compatible. When it is not, the mismatch corresponds to a nontrivial predictive defect at the chosen tolerance.

Thus the experiments provide empirical access to overlap compatibility, not to a nontrivial cohomology class. They motivate studying local test coverage and global compatibility jointly, while leaving the causal explanation of their relationship open.

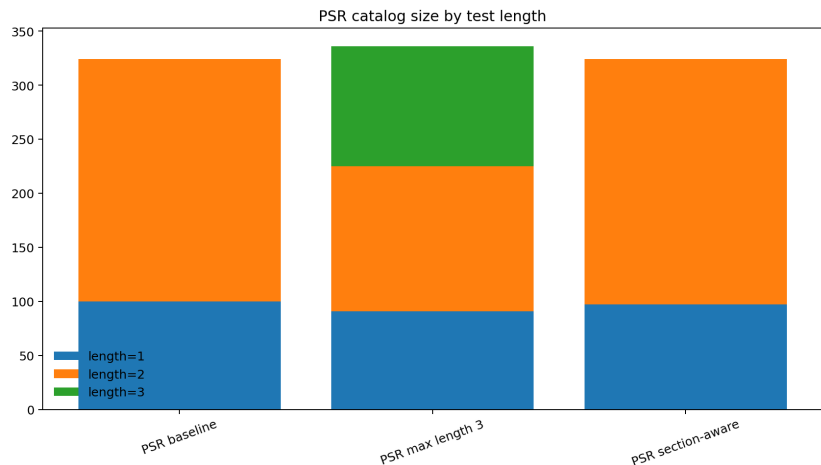


Figure 59: PSR test catalog size decomposed by test length. Increasing test length expands the catalog and increases structural coverage, while section-aware extraction redistributes test mass toward shorter, more context-specific patterns.

The below figure makes the structural effect of longer tests explicit. Increasing the maximum test length shifts mass toward longer compositional patterns, enlarging the predictive basis available in each local context. In the language of Theorem 21, this corresponds to increasing a catalog-size proxy, not an estimated local predictive rank.

However, the predictive gains remain modest. This suggests that most of the low-order predictive structure is already captured by short tests, and that additional tests primarily improve representational completeness rather than discriminative power.

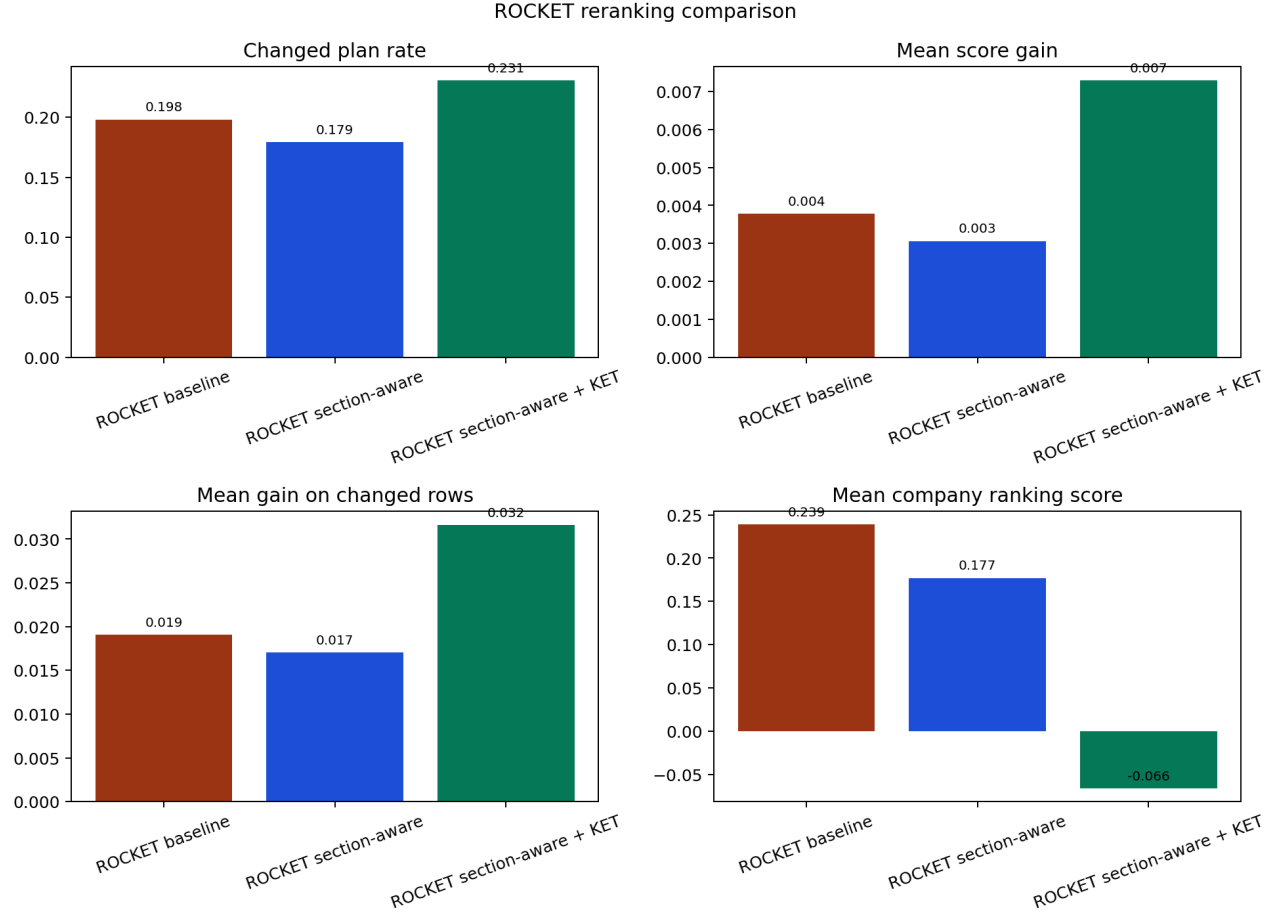
The below figure highlights a complementary phenomenon on the ROCKET side. The section-aware + KET variant produces the largest change rate and the largest mean score gains at the workflow level, indicating that it performs the most aggressive local transformations. However, these gains do not translate into uniformly improved company-level rankings.

One possible explanation is that stronger local transformations increase overlap mismatches across contexts. The present ROCKET scores do not measure δ_{ij} , however, so this remains a hypothesis rather than an inference from the figure.

Taken together, the PSR and ROCKET plots provide a first descriptive calibration. Increasing test length enlarges the catalog without materially changing the reported predictive metrics, while aggressive workflow reranking

Table 53: Representative compressed PSR tests.

Test	Variant	Support	$p(\text{true})$	$p(\text{false})$	Gap	Brier
<code>realize_revenue => margin_up</code>	Baseline	588	0.366	0.263	0.103	0.175
<code>optimize -> expand -> realize_revenue => margin_up</code>	Max length 3	548	0.380	0.270	0.110	0.177
<code>expand -> market -> realize_revenue => margin_up</code>	Max length 3	503	0.370	0.262	0.108	0.174
<code>price -> realize_revenue => margin_up</code>	Section-aware	148	0.379	0.271	0.108	0.176
<code>acquire -> expand => revenue_up</code>	Section-aware	58	0.297	0.325	-0.028	0.224



does not yield uniformly positive company scores. Neither result estimates predictive rank or a cohomological obstruction.

Temporal Structure of Predictive Obstruction

Definition 87 (Predictive obstruction index). *Let $\{U_i\}_{i \in I}$ be a finite cover of contexts, and let $s_i \in \mathcal{F}(U_i)$ be local predictive sections. For each overlap $U_i \cap U_j$, define the mismatch*

$$\delta_{ij} = \rho_{U_i, U_i \cap U_j}(s_i) - \rho_{U_j, U_i \cap U_j}(s_j).$$

Fix a norm $\|\cdot\|_{ij}$ on $\mathcal{F}(U_i \cap U_j)$. The predictive obstruction index is

$$\text{POI}(\{s_i\}) = \left(\frac{1}{|E|} \sum_{(i,j) \in E} \|\delta_{ij}\|_{ij}^2 \right)^{1/2},$$

where E is the set of overlapping context pairs.

Figure 60: ROCKET reranking comparison across variants. The section-aware + KET model produces the largest structural changes and score gains, but does not yield uniformly positive company-level ranking scores. This illustrates a possible tension between aggressive local optimization and global consistency.

INTERPRETATION. The predictive obstruction index measures the extent to which local predictive states fail to agree on overlaps. A value of zero means exact gluing at the level of local sections, while larger values indicate increasing incompatibility between local predictive descriptions. In empirical terms, it quantifies the degree of inconsistency between predictive stories extracted from different operational contexts of the same firm.

Definition 88 (Stable aggregation hypothesis). *A chosen reconstruction scheme satisfies stable aggregation with constant C if, for every admitted family of local sections, it constructs a global candidate $\bar{s}$ satisfying*

$$\frac{1}{|I|} \sum_{i \in I} \|\rho_i(\bar{s}) - s_i\|_i \leq C \text{POI}(\{s_i\}).$$

This is an analytic assumption on the cover, norms, restriction maps, and reconstruction operator. Pairwise-small mismatches alone do not imply it in an arbitrary presheaf.

Theorem 23 (Predictive error bound under stable aggregation). *Let $\{U_i\}_{i \in I}$ be a finite context cover, and let $s_i \in \mathcal{F}(U_i)$ be local predictive sections with local prediction losses*

$$\mathcal{L}_i(s_i).$$

Assume that each loss is Lipschitz with respect to the predictive norm:

$$|\mathcal{L}_i(f) - \mathcal{L}_i(g)| \leq L_i \|f - g\|_i.$$

Let $\bar{s}$ be an approximate global predictor constructed from the local sections, and assume the stable aggregation hypothesis above holds with constant C .

Then

$$\mathcal{L}(\bar{s}) \leq \frac{1}{|I|} \sum_{i \in I} \mathcal{L}_i(s_i) + L C \text{POI}(\{s_i\}),$$

where $L = \max_i L_i$ and C depends only on the cover and the gluing scheme.

No converse lower bound follows from the categorical data alone. Such a bound would require a stable-separation inequality for the particular cover and norms; without it, overlap mismatches may contain null directions that do not control global predictive loss.

Proof sketch of Theorem 23. Apply the stable aggregation hypothesis and Lipschitz continuity of the local losses. Averaging the pointwise Lipschitz inequalities gives

$$\frac{1}{|I|} \sum_i \mathcal{L}_i(\rho_i(\bar{s})) \leq \frac{1}{|I|} \sum_i \mathcal{L}_i(s_i) + L C \text{POI}(\{s_i\}).$$

Averaging over contexts yields the result. □

INTERPRETATION IN THE 10-K DOMAIN. In the corporate filing setting, the predictive obstruction index measures how strongly local predictive models extracted from different sections of a firm’s disclosures disagree on shared workflow transitions. For example, an operations context may assign high predictive support to

$$\text{optimize} \rightarrow \text{expand} \Rightarrow \text{revenue\_up},$$

while a market context assigns low support to the same overlap test. A high mismatch index therefore indicates disagreement among the selected local predictors. Under the stable aggregation and Lipschitz assumptions, Theorem 23 upper-bounds one contribution to predictive error.

Definition 89 (Empirical predictive obstruction index). *For a company-year, let Δ_{ij} be the maximum absolute mismatch of overlap probabilities between contexts U_i and U_j . Define*

$$\text{POI}_{\text{emp}} = \frac{1}{|E|} \sum_{(i,j) \in E} \Delta_{ij} \quad \text{or} \quad \text{POI}_{\text{emp}}^{\max} = \max_{(i,j) \in E} \Delta_{ij}.$$

Definition 90 (Empirical glueability threshold). *If*

$$\text{POI}_{\text{emp}}^{\max} \leq \tau,$$

for a chosen tolerance τ , the implementation labels the company-year empirically glueable. This is a decision rule. It does not imply that expected global error increases monotonically above τ ; that relationship must be estimated and depends on the stable aggregation and loss assumptions.

INTUITION: The predictive mismatch index measures how hard it is for the chosen reconstruction scheme to reconcile local predictive stories. A small value means approximate agreement on the measured overlaps. Accuracy, and any unavoidable penalty for a large value, require the additional analytic and empirical assumptions stated above.

RELATION TO COHOMOLOGY. The empirical index measures the norm of the mismatch coboundary. It is a useful approximate-descent statistic, but it is not the norm of a nontrivial cohomology class. A cohomological formulation would require a separate extension problem whose cocycle is not exact by construction.

These plots provide a temporal view of two implementation-level statistics. The incomparability ratio was proposed as a proxy for mismatch, while the success composite aggregates downstream indicators. No calibration experiment here establishes that incomparability estimates the predictive mismatch defined above.

Incomparability varies over time and across these companies, and its displayed association with the success composite is not monotonic. These

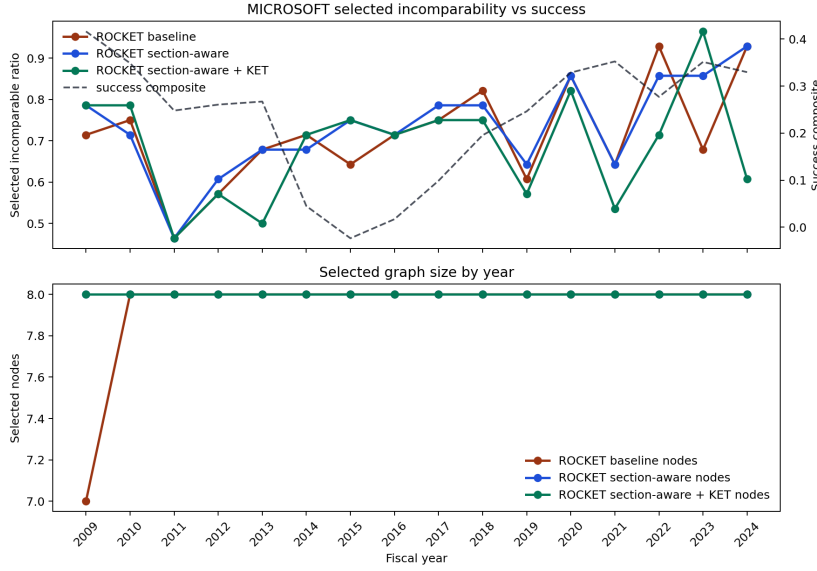


Figure 61: Microsoft: the implementation's incomparability statistic and success composite over time. Different variants produce different descriptive profiles; the plot does not validate incomparability as a predictive-mismatch measure.

observations describe the plots only; they do not show that local diversity is beneficial or that weak outcomes are caused by a breakdown of coherence.

Theorem 21 supplies a possible interpretation only after a link between incomparability and the overlap mismatches δ_{ij} has been validated. Establishing that link, and then testing whether firms move between coherent and incoherent predictive regimes, remains an empirical question.

Definition 91 (Panel obstruction distribution). *Let $\mathcal{Z}$ denote the set of company-year instances in the panel, equipped with an empirical distribution $\mathbb{P}_{\text{panel}}$. For each $z \in \mathcal{Z}$, let $\text{POI}(z)$ be its predictive obstruction index. The panel obstruction distribution is the pushforward measure*

$$\mu_{\text{POI}} = (\text{POI})_{\#} \mathbb{P}_{\text{panel}},$$

that is, the distribution of obstruction values across the panel.

Definition 92 (Panel obstruction summary statistics). *Given μ_{POI} , define:*

$$\bar{O} = \mathbb{E}[\text{POI}], \quad \sigma_O^2 = \text{Var}(\text{POI}), \quad G_{\tau} = \mathbb{P}(\text{POI} \leq \tau).$$

Here G_{τ} is the panel glueability rate at tolerance τ .

Theorem 24 (Panel error bound via obstruction distribution). *Let $\mathbb{P}_{\text{panel}}$ be the empirical distribution over company-year instances, and let $\text{POI}(z)$ denote the predictive obstruction index of instance z . Suppose that for each z there exist local predictive sections $\{s_i^{(z)}\}$ over a finite context cover, and let $\bar{s}^{(z)}$ be an approximate global predictor constructed from these local sections.*

Assume:

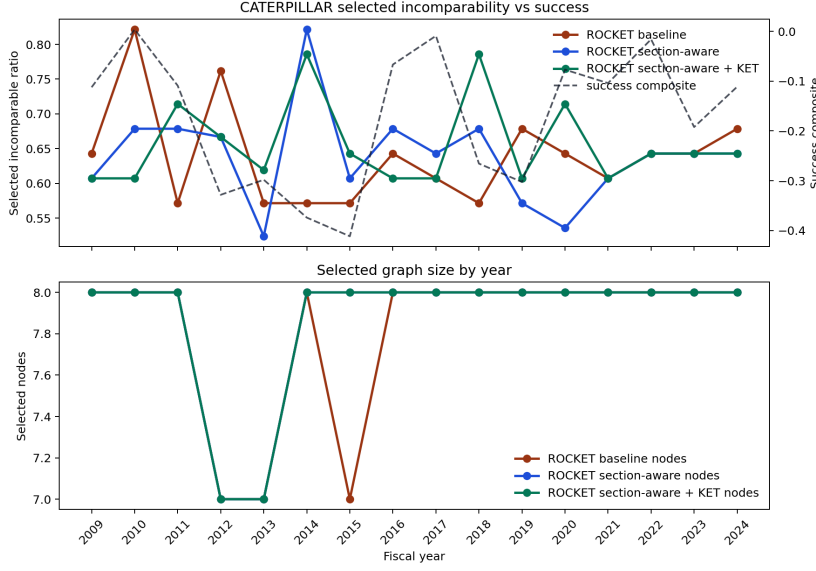


Figure 62: Caterpillar: temporal variation in the implementation's incomparability statistic and success composite. The association is descriptive and does not identify sector-specific dynamics.

1. *local losses are uniformly Lipschitz:*

$$|\mathcal{L}_i(f) - \mathcal{L}_i(g)| \leq L\|f - g\|_i;$$

2. *approximate gluing holds with constant C :*

$$\|\rho_i(\bar{s}^{(z)}) - s_i^{(z)}\|_i \leq C \text{POI}(z);$$

3. *the panel risk is defined by*

$$\mathcal{R}(\bar{s}) = \mathbb{E}_{z \sim \mathbb{P}_{\text{panel}}} \left[\frac{1}{|I_z|} \sum_{i \in I_z} \mathcal{L}_i(\rho_i(\bar{s}^{(z)})) \right].$$

Then

$$\mathcal{R}(\bar{s}) \leq \mathbb{E}_{z \sim \mathbb{P}_{\text{panel}}} \left[\frac{1}{|I_z|} \sum_{i \in I_z} \mathcal{L}_i(s_i^{(z)}) \right] + LC \mathbb{E}_{z \sim \mathbb{P}_{\text{panel}}} [\text{POI}(z)].$$

Equivalently,

$$\mathcal{R}(\bar{s}) \leq \mathcal{R}_{\text{local}} + LC \bar{O}.$$

Proof sketch. Apply the single-instance obstruction bound to each company-year instance z :

$$\frac{1}{|I_z|} \sum_{i \in I_z} \mathcal{L}_i(\rho_i(\bar{s}^{(z)})) \leq \frac{1}{|I_z|} \sum_{i \in I_z} \mathcal{L}_i(s_i^{(z)}) + LC \text{POI}(z).$$

Taking expectation over the panel distribution yields the result. $\square$

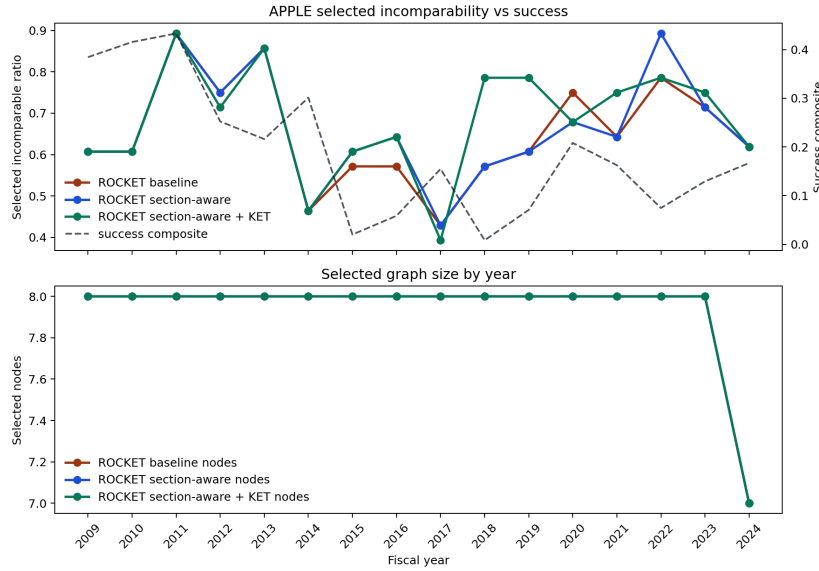


Figure 63: Apple: incomparability and success composites across variants. The plot records co-variation in constructed statistics, not a causal effect of local or global coherence on performance.

Corollary 1 (Tail-sensitive panel bound). *For any threshold $\tau > 0$,*

$$\mathcal{R}(\bar{s}) \leq \mathcal{R}_{\text{local}} + LC \tau + LC \mathbb{E}[\text{POI} \mathbf{1}_{\{\text{POI} > \tau\}}].$$

Hence the contribution of obstruction decomposes into a glueable part and a tail penalty from non-glueable company-years.

Proposition 6 (Comparison of obstruction upper bounds). *Let v_1 and v_2 be two workflow-selection variants, with local risks $\mathcal{R}_{\text{local}}^{(v_1)}$, $\mathcal{R}_{\text{local}}^{(v_2)}$ and mean mismatch values $\bar{O}^{(v_1)}$, $\bar{O}^{(v_2)}$. Define the certified upper bound*

$$B(v) = \mathcal{R}_{\text{local}}^{(v)} + LC \bar{O}^{(v)}.$$

Then

$$B(v_2) - B(v_1) = (\mathcal{R}_{\text{local}}^{(v_2)} - \mathcal{R}_{\text{local}}^{(v_1)}) + LC(\bar{O}^{(v_2)} - \bar{O}^{(v_1)}).$$

This compares the two upper bounds, not the unknown risk difference $\mathcal{R}^{(v_2)} - \mathcal{R}^{(v_1)}$; subtracting two one-sided risk bounds would not justify such an inequality.

INTERPRETATION IN THE 10-K PANEL. In the corporate filing setting, each company-year contributes a local family of predictive sections derived from contexts such as operations, market, finance, and innovation. The predictive obstruction index measures how strongly these sections disagree on shared workflow transitions. The panel obstruction distribution records how such incompatibilities are distributed across the entire 10-K panel.

Theorem 24 gives a conditional upper bound with two terms: the average quality of local predictive models and the mean mismatch across company-years. The bound motivates reporting both, but does not identify which term causes the heterogeneous company scores.

Definition 93 (Catalog-complexity saturation). *Let $\mathcal{R}(m)$ denote the panel-level predictive risk of a PSR model built from tests of maximum length m , and let $\hat{d}(m)$ denote a declared complexity proxy such as catalog size or mean supported tests per state. It is called a rank only if it is computed from the corresponding Hankel operator.*

We say that the measured catalog exhibits complexity saturation beyond m_0 if:

1. $\hat{d}(m)$ continues to increase for $m \geq m_0$, but
2. the incremental predictive gain becomes negligible, in the sense that

$$|\mathcal{R}(m+1) - \mathcal{R}(m)| \leq \varepsilon$$

for all $m \geq m_0$ and some small tolerance $\varepsilon > 0$.

Definition 94 (Metric saturation). *Let $M(m)$ be a predictive quality metric, such as AUC or negative Brier score. We say that the metric saturates beyond m_0 if $\hat{d}(m)$ continues to increase for $m \geq m_0$ while*

$$|M(m+1) - M(m)| \leq \varepsilon$$

for all $m \geq m_0$.

Remark 25 (Predictive rank saturation heuristic). *Suppose that for a given context cover, the local predictive systems admit finite effective dimensions d_U^* , and that the dominant predictive dependencies are captured by tests of length at most m_0 .*

Under a separate assumption that the residual contribution η_m of longer tests tends to zero, one expects for $m \geq m_0$:

1. the empirical predictive basis $\hat{d}(m)$ may continue to grow;
2. the obstruction index may weakly decrease due to improved overlap coverage;
3. but the panel-level predictive risk satisfies

$$\mathcal{R}(m) = \mathcal{R}(m_0) + O(\eta_m),$$

where $\eta_m \rightarrow 0$ measures the residual predictive contribution of higher-order tests.

This is a restatement of the residual-contribution assumption, not an independent theorem about PSRs.

Remark 26 (Saturation under bounded mismatch improvement). *Assume the hypotheses of Theorem 24. Suppose that for $m \geq m_0$ the local predictive risk has saturated:*

$$\mathcal{R}_{\text{local}}(m) = \mathcal{R}_{\text{local}}(m_0) + O(\eta_m),$$

and that the obstruction improvement is bounded:

$$\bar{O}(m_0) - \bar{O}(m) \leq \gamma_m,$$

with $\eta_m, \gamma_m \rightarrow 0$.

If, in addition, matching lower control or a direct risk estimate establishes that the upper bound is tight to the same order, then

$$\mathcal{R}(m) = \mathcal{R}(m_0) + O(\eta_m + \gamma_m).$$

The one-sided upper bound alone does not imply this equality or prove vanishing returns.

EMPIRICAL EVIDENCE FOR SATURATION. Predictive rank saturation means that local predictive state has become effectively sufficient before the full combinatorial test space has been exhausted. The PSR experiments show a pattern consistent with catalog saturation, not a measured saturation of predictive rank. Increasing the maximum test length from 2 to 3 increases both the catalog size and the number of supported tests per state, and even improves the empirical gluing rate to 588/588. However, the predictive gains are minimal: the Brier score is essentially unchanged and the AUC improves only slightly. Because only one increment in maximum length is tested and the bootstrap intervals overlap, this suggests—but does not establish—that short tests capture most of the predictive signal in this construction.

Summary

These results provide an implementation-level illustration of the chapter. The BASKET–ROCKET pipeline can be lifted into a sheaf-theoretic predictive-state layer that is both computationally realizable and empirically inspectable. Local contexts, restriction maps, and overlap mismatches are not abstract constructs but concrete objects computed from real 10-K data.

Local predictive states can be estimated and overlap compatibility can be tested empirically at a declared tolerance. The present experiment does not measure a nontrivial cohomology class or prove hidden-state recovery. It provides a bridge between categorical semantics and real-world data analysis in agentic systems.

Suggested Reading

Predictive representations of state were introduced by [Littman et al. \[2001\]](#) and developed into a general dynamical-systems theory by [Singh et al.](#)

[2004]. The coalgebraic language of states and observations is developed by Jacobs [2016] and Rutten [2000]; Mac Lane and Moerdijk [1992] supplies the local-to-global machinery used in the topos formulation.

Causal Density Functions

In this chapter we study *causal density functions*: ordinary Radon–Nikodym density ratios comparing a specified intervention or perturbation law with a baseline observational law. The construction is a change-of-measure device, not an identification theorem. It is meaningful only after the target law has been identified or directly sampled and only when the required domination condition holds. We use the ratios for calibration diagnostics, dominated regime-response curves, and an exploratory pairwise score whose limitations are visible in the experiments.

Introduction

Causal inference distinguishes observational conditioning from interventions⁴⁵. A Radon–Nikodym derivative does not create that distinction and does not turn an observational law into an interventional one. Instead, if a target law P_a has already been specified and satisfies $P_a \ll P_0$, the derivative dP_a/dP_0 transports integrals under P_0 to integrals under P_a . This classical identity is useful but demanding: hard interventions on continuous coordinates can be singular, and weak empirical overlap can make estimated weights unstable even when population absolute continuity holds.

The chapter therefore separates four questions that were previously conflated: identification of the target interventional law, existence of the density ratio, statistical estimation of that ratio, and the validity of any downstream causal-discovery score. No Kan-extension duality, adjunction, or Beck–Chevalley statement is needed for these claims.

Causal Density Functions

The construction requires two probability laws on the same measurable outcome space. It does not supply either law or the assumptions needed to identify an interventional law from observational data.

Definition 95 (Causal density function). *Let $Z = (X_1, \dots, X_d)$ take values in a measurable space $(\mathcal{Z}, \mathcal{A})$. Let P_0 be an observational law and let $\{P_a : a \in \mathcal{I}\}$ be a specified family of intervention or perturbation laws on that*

⁴⁵ Guido W. Imbens and Donald B. Rubin. *Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction*. Cambridge University Press, USA, 2015. ISBN 0521885884; and Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, USA, 2nd edition, 2009. ISBN 052189560X

same space. If $P_a \ll P_0$, the causal density function for regime a is

$$\rho_a(z) = \frac{dP_a}{dP_0}(z).$$

IDENTIFICATION AND DOMINATION. The adjective “causal” is justified only after P_a has been identified. With randomized perturbation samples, P_a is the empirical regime law. With observational data alone, assumptions such as consistency, positivity and conditional exchangeability, or a valid do-calculus identification formula, must come first. A deterministic hard intervention on a continuously distributed coordinate is commonly singular relative to the observational joint law, so the joint ratio is then undefined.

CALIBRATION IDENTITY. For every integrable statistic f , the defining property is

$$\mathbb{E}_a[f(Z)] = \mathbb{E}_0[f(Z)\rho_a(Z)]. \quad (43)$$

For an estimated ratio, small held-out gaps for finitely many choices of f diagnose only those transported moments; they do not prove equality of the full distributions.

USES AND SCOPE. We use causal densities for calibration, response curves over dominated soft or stochastic intervention families, and exploratory directed scores. The pairwise residual score is not a generally consistent structure-learning procedure: it can confuse direct effects, mediated associations, common causes, and multiple-parent structure. Collecting the ratios into $\rho(z) = (\rho_a(z) : a \in \mathcal{I})$ is only organizational notation, not a claim that they form a natural transformation.

EXAMPLE (FINITE CASE). For environments e_0 (observational) and e_1 (interventional) with empirical distributions $p_0(x)$ and $p_1(x)$ on a finite set X ,

$$\rho(x) = \frac{p_1(x)}{p_0(x)}, \quad s_{ij} = - \sum_x \rho(x) (x_j - \hat{x}_j)^2 / \text{Var}(X_j).$$

The first equation is the finite change-of-measure ratio. The second is an example of a ratio-weighted residual score; it is a modeling choice, not a consequence of the Radon–Nikodym theorem.

Analytic Distinctions and Statistical Scope

INTERVENTION RATIO VERSUS DEPENDENCE RATIO. The intervention ratio dP_a/dP_0 must not be confused with the observational dependence density

$$\frac{dP_{XY}}{d(P_X \otimes P_Y)}(x, y).$$

The latter equals one under independence and enters mutual information, but it is neither directed nor causal without additional structure.

FINITE REGIME CHANGES VERSUS INFINITESIMAL SCORES. For a differentiable path of dominated densities, the score is $s_\lambda(z) = \partial_\lambda \log p_\lambda(z)$, and

$$\log \frac{p_{\lambda+\delta}(z)}{p_\lambda(z)} = \delta s_\lambda(z) + o(\delta).$$

This local expansion must not be identified with a finite intervention ratio or with the derivative of an observational dependence density.

WHAT RATIO CONSISTENCY REQUIRES. Pointwise convergence of fitted log-densities does not imply $L^1(P_0)$ convergence of their ratio: a fitted denominator can approach zero on shrinking sets and create arbitrarily large weights. A sufficient route is convergence $\hat{\rho}_n \rightarrow \rho$ in P_0 -probability together with uniform integrability of $\{\hat{\rho}_n\}$; Vitali's theorem then yields

$$\|\hat{\rho}_n - \rho\|_{L^1(P_0)} \longrightarrow 0.$$

WHAT IS NOT CLAIMED. We make no categorical universal-property claim for the Radon–Nikodym derivative. In particular, Kan extensions, Beck–Chevalley conditions, and the previously proposed “RN–Kan duality” do not identify P_a , estimate the ratio, or provide a consistency theorem.

Estimating Causal Density Functions

TWO-DENSITY PLUG-IN. For a regime a , fit densities $\hat{p}_0$ and $\hat{p}_a$ on a shared support and form

$$\hat{\rho}_a(z) = \exp(\log \hat{p}_a(z) - \log \hat{p}_0(z)).$$

This is a baseline rather than a uniquely preferred estimator. Error in both density estimates enters the quotient, so direct or classifier-based ratio estimators, inverse-probability methods, outcome regression, Riesz representers, or doubly robust estimators may be preferable for particular identified targets.

HELD-OUT CALIBRATION. For prespecified f , report both the weighted and unweighted gaps

$$\hat{\Delta}_f^w(a) = \left| \hat{\mathbb{E}}_0[f(Z)\hat{\rho}_a(Z)] - \hat{\mathbb{E}}_a[f(Z)] \right|, \quad \hat{\Delta}_f^0(a) = \left| \hat{\mathbb{E}}_0[f(Z)] - \hat{\mathbb{E}}_a[f(Z)] \right|.$$

Evaluation should use held-out data and quantify uncertainty with repeated splits or bootstrap intervals.

Algorithm 17: Causal Density Estimation and Scoring

Input: Observational samples, regime samples, variables $X_1, \dots, X_d$

- 1: **for** each intervention or perturbation regime a **do**
 - 2: Fit $\hat{p}_0$ and $\hat{p}_a$ on a shared support.
 - 3: Compute $\hat{\rho}_a = \exp(\log \hat{p}_a - \log \hat{p}_0)$.
 - 4: Evaluate held-out weighted and unweighted gaps for selected f 's.
 - 5: **end for**
 - 6: **for** each ordered pair $X_i \rightarrow X_j$ **do**
 - 7: Fit $\hat{\mathbb{E}}[X_j \mid X_i]$ and compute s_{ij} .
 - 8: **end for**
 - 9: Return ratio estimates, diagnostics, response curves, and optional exploratory scores.
-

EXPLORATORY DIRECTED SCORE. For an ordered pair $X_i \rightarrow X_j$, the experiments use

$$s_{ij} = \mathbb{E}_0 \left[\hat{\rho}_i(Z) \frac{(X_j - \hat{\mathbb{E}}[X_j \mid X_i])^2}{\text{Var}(X_j)} \right].$$

Lower values indicate only a better pairwise fit under this weighted criterion. Sparse selection and cycle pruning do not repair omitted-variable or mediation bias.

Proposition 7 (A sufficient transport condition). *Suppose $\hat{\rho}_{a,n} \rightarrow \rho_a$ in P_0 -probability and $\{\hat{\rho}_{a,n}\}$ is uniformly integrable under P_0 . Then $\|\hat{\rho}_{a,n} - \rho_a\|_{L^1(P_0)} \rightarrow 0$. Consequently, for every bounded measurable f ,*

$$|\mathbb{E}_0[f(Z)\hat{\rho}_{a,n}(Z)] - \mathbb{E}_a[f(Z)]| \rightarrow 0.$$

Proof. Vitali's convergence theorem gives $L^1(P_0)$ convergence. The second claim follows from the Radon–Nikodym identity and the bound $\|f\|_\infty \|\hat{\rho}_{a,n} - \rho_a\|_1$. $\square$

This proposition assumes ratio convergence; it does not assert that normalizing-flow maximum likelihood automatically supplies it.

Experimental Results

We now describe a suite of experiments testing Kan-Do-Calculus on extracting causal structure from numerical data, in the long tradition of work on causal discovery (see ⁴⁶ for a detailed survey). Some experimental results are in the main chapter, and the remainder are detailed in the Supplementary Materials.

⁴⁶ Alessio Zanga and Fabio Stella. A survey on causal discovery: Theory and practice, 2023. URL <https://arxiv.org/abs/2305.10032>

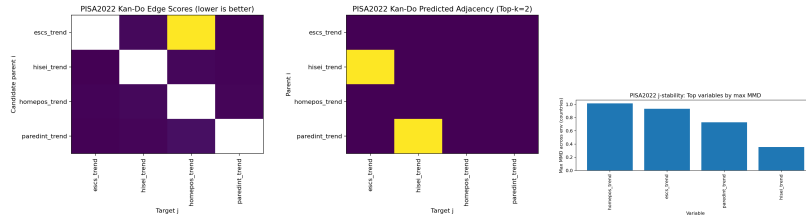


Figure 64: **PISA 2022 causal-density diagnostics.** Left: ratio-weighted pairwise scores (s_{ij} ; lower is better). Middle: predicted adjacency (Top- $k = 2$). Right: cross-country j -stability (max MMD).

PISA 2022 Socio-Economic Panel

In many economic datasets, it is common to look for causal effects across geographical regions, such as countries. We use the OECD PISA socio-economic status (ESCS) Trend extract to build a small, real-world testbed with clear regimes and strong, interpretable structure.⁴⁷ We evaluate the causal-density baseline on a four-variable socio-economic panel constructed from PISA 2022. Each row corresponds to a country and each column is a standardized index used in OECD’s socio-economic measurements: `escs_trend` (economic, social, cultural status), `hisei_trend` (highest international socio-economic index), `homepos_trend` (home possessions), and `paredint_trend` (parental education). The column `env` identifies the environment (country), yielding a multi-regime dataset analogous to the perturbation regimes in S9 and LINC5 experiments reported below.

Experiment: PISA causal-density diagnostic

The figure displays the ratio-weighted pairwise scores, predicted adjacency, and cross-country j -stability (maximum MMD per variable). The lowest-scoring pairs include `paredint_trend` → `hisei_trend` and `hisei_trend` → `escs_trend`, but the panel alone does not identify these arrows causally. The j -stability bar plot highlights substantial cross-country heterogeneity for variables like `homepos_trend`, while `hisei_trend` remains the most stable indicator. Low MMD is reported as a cross-regime stability diagnostic; it is not by itself a certificate of causal identification or correct graph recovery.

DO-CURVE FOR `HISEI_TREND` → `ESCS_TREND`. To illustrate density reweighting in a socio-economic context, we compute a regime-response curve for the pair (`hisei_trend`, `escs_trend`) using a two-dimensional RN-flow. The variable `hisei_trend` is a standardized socio-economic index used internationally in PISA, while `escs_trend` aggregates broader economic, social, and cultural status components. For a grid of values x , we construct the soft intervention $\text{do}(\text{hisei\_trend} = x)$ via the Radon-Nikodym density ratio, and evaluate the corresponding expected value of `escs_trend`.

⁴⁷ The PISA datasets are available at <https://webfs.oecd.org/pisa2022/index.html>.

The figure displays the resulting do-curve. Despite substantial cross-country heterogeneity, the interventional response is globally monotone: increasing `hisei_trend` produces a steady increase in expected `escs_trend`. Notably, the curve exhibits plateaus and nonlinear transitions, reflecting the discretized and thresholded nature of PISA socio-economic indices. This example shows the response curve produced by the fitted ratio. Given the large calibration error and unclear intervention semantics of country regimes, it should not be read as an identified policy effect.

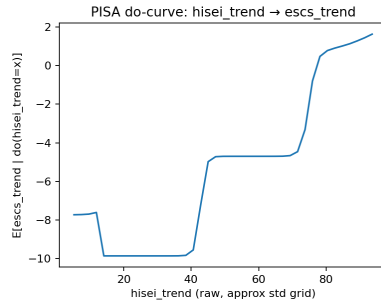


Figure 65: **PISA do-curve: hisei_trend → escs_trend**. Response computed via fitted density-ratio reweighting. The relationship is globally monotone with nonlinear plateaus, consistent with the ordinal structure of PISA socio-economic indices.

RN CALIBRATION FOR `HISEI_TREND` → `ESCS_TREND`. To complete the cross-dataset comparison of Radon–Nikodym calibration, we evaluate the RN identity on the PISA2022 pair (`hisei_trend`, `escs_trend`), using the same two-dimensional RN-flow as in the do-curve computation. The figure reports the absolute gaps

$$|\mathbb{E}_{\text{obs}}[f(Y)\rho] - \mathbb{E}_{\text{do}}[f(Y)]| \quad \text{for } f(y) = y, y^2.$$

The calibration errors are substantially larger than in the S9 and LINC domains (approximately 4.0 for $f(y) = y$ and 20.0 for $f(y) = y^2$). This reflects the extreme heterogeneity across countries: socio-economic indices such as `hisei_trend` and `escs_trend` have widely separated distributions across environments, and the RN density ratio must transport probability mass across non-overlapping regimes. The large errors are consistent with weak empirical overlap and substantial cross-regime heterogeneity. This is a failure diagnostic for the fitted ratio, not evidence for a categorical gluing theorem.

Sachs Protein Signaling

We evaluate Kan-Do-Calculus on the classical *Sachs* flow-cytometry dataset of protein-signaling pathways in human immune cells (11 phospho-proteins/phospho-lipids, single-cell measurements). Each stimulation or inhibitor setting defines an experimental regime (e.g., `CD3CD28`, `PKA_inh`, `PKC_act`, ...); we encode these as an environment label `env`. Following the standard public benchmark⁴⁸, we treat each condition as a soft or hard intervention on a subset of signaling nodes:

⁴⁸ Karen Sachs, Omar Perez, Dana Pe’er, Douglas A. Lauffenburger, and Garry P. Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, 2005. DOI: 10.1126/science.1105809

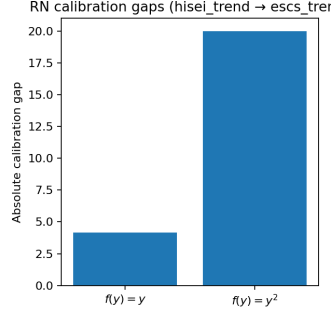


Figure 66: **RN calibration for $\text{hisei\_trend} \rightarrow \text{escs\_trend}$.** Large calibration gaps reflect the substantial cross-country divergence of PISA socio-economic indices and warn against interpreting the fitted response curve causally.

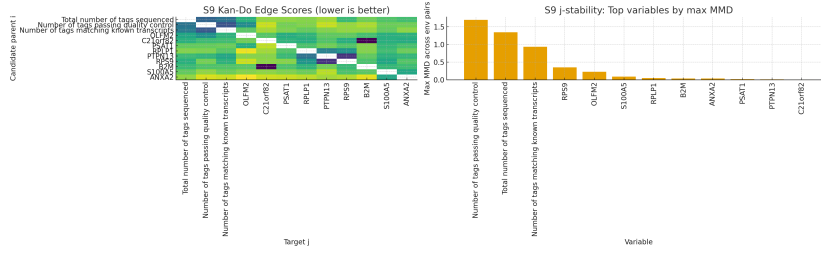


Figure 67: **Sachs causal-density results.** Left: causal-density edge-score heatmap (s_{ij} ; lower is better). Right: regime divergence (max MMD per variable). The heatmap is exploratory; the corresponding pairwise graph has F1 0.1176.

{raf, mek, erk, pka, pkc, pip2, pip3, plcγ, akt, p38, jnk}.

RESULTS. The figure visualizes the resulting causal-density edge-score matrix and the per-variable regime-divergence statistics. Against the 12-edge reference graph in the released evaluation, the pairwise scorer has SHD 30, F1 0.1176, precision 0.0909, and recall 0.1667 (TP 2, FP 20, FN 10). The regime-penalized variant has SHD 16 and F1 0.1111 (TP 1, FP 5, FN 11). Some canonical directions receive favorable scores, but the aggregate results do not support a structure-learning claim.

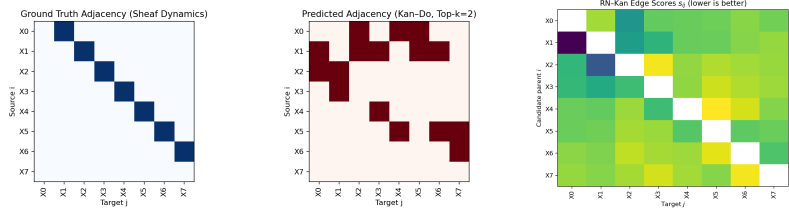
Multi-Regime Chain Experiment

To test causal-density scoring when multiple regimes are available, we construct a synthetic dynamical system in which each regime represents a local perturbation of an underlying causal process. The domain consists of $d = 8$ scalar variables forming a linear chain $X_1 \rightarrow X_2 \rightarrow \dots \rightarrow X_8$. Within each regime $r \in \{1, \dots, 4\}$, the mechanism for each variable is given by

$$X_j^{(r)} = 0.8 X_{j-1}^{(r)} + \epsilon_j^{(r)}, \quad \epsilon_j^{(r)} \sim \mathcal{N}(0, 0.4^2),$$

with soft interventions applied to a small subset of nodes by shifting their means (e.g., $X_1 + 0.8$ in regime 2 and $X_2 - 0.6$ in regime 3). This generates a set of local perturbation models with a known chain ground truth.

The figure illustrates the resulting adjacency matrices and edge-score field. The Top- $k = 2$ pairwise graph contains several extra parents. The



aggregate result reported in Table 55 is $\text{SHD} = 8$ and $\text{F1} = 0.33$, so the visible low-score band near the chain is not evidence of near-perfect recovery. Regime-stability penalties can alter the selected graph, but this experiment does not establish a general consistency result.

The implementation also reports maximum mean discrepancy across regimes. These statistics describe selected distributional differences; they neither establish a sheaf gluing condition nor imply recovery of the true graph.

CROSS-DOMAIN BEHAVIOR OF RN CALIBRATION. The table reports values in the released CSV artifacts. The synthetic first-moment gap is 2.023, the LINCS gaps are 0.127 and 0.322, and the PISA gaps are 4.145 and 20.005. No released file supports previously quoted Sachs gaps, so those values are omitted. Without unweighted gaps, cross-fitting, or uncertainty intervals, these numbers audit plug-in outputs but do not show that weighting improves transport or isolate why the errors differ.

Discussion: Causal Density and Sheaf Coherence

The S9 score matrix ranks some canonical signaling directions highly, but the recovered graph has only $\text{F1} = 0.12$. The multi-regime chain reaches $\text{F1} = 0.33$. These results do not show that the pairwise score identifies stable mechanisms or that causal consistency is equivalent to sheaf consistency. The useful lesson is methodological: moment transport, cross-regime stability, and graph recovery are distinct properties and must be evaluated separately.

Dataset / Pair	$ \Delta_{f(y)=y} $	$ \Delta_{f(y)=y^2} $
Synthetic Gaussian	2.023	0.107
LINCS	0.127	0.322
(HSPA8 $\rightarrow$ CDC25B)		
PISA2022	4.145	20.005
(hisei_trend $\rightarrow$ escs_trend)		

Figure 68: **Multi-regime chain pairwise density-ratio baseline.** Left: Ground-truth chain structure ($X_0 \rightarrow X_1 \rightarrow \dots \rightarrow X_7$). Middle: Predicted adjacency from the fast Kan-Do model (Top- $k = 2$). Right: ratio-weighted edge-score matrix s_{ij} (lower is better). Extra edges remain and the aggregate recovery is only $\text{SHD} = 8$, $\text{F1} = 0.33$.

Table 54: **RN calibration gaps across domains.** Absolute errors between $\mathbb{E}_{\text{obs}}[f(Y)\rho]$ and $\mathbb{E}_{\text{do}}[f(Y)]$ for $f(y) = y$ and $f(y) = y^2$. Smaller is better. The entries reproduce the released calibration CSVs. No corresponding Sachs calibration artifact was found.

Dataset	SHD ↓	F1 ↑
Synthetic DAG (ER, d=10)	37	0.10
Multi-regime chain	8	0.33
Sachs (pairwise)	30	0.12
Sachs (multi + regime penalty)	16	0.11

Table 55: **Directed structure recovery for causal-density edge scoring.**

Synthetic Benchmark Experiments

To complement the real-world experiments (Sachs, LINCS, PISA) and the multi-regime sheaf example, we evaluate Kan-Do-Calculus on standard synthetic causal discovery benchmarks. These synthetic tests use linear-Gaussian Erdős-Rényi DAGs, following settings used by NOTEARS, DCDI, CAM, and related methods. The table reports the directed-structure metrics for the evaluated pairwise scorer.

RESULTS. For the evaluated instance with $d = 10$ nodes and $p = 0.2$, the pairwise scorer achieves

$$\text{SHD} = 37, \quad \text{F1} = 0.10.$$

This single synthetic result is a negative calibration of the pairwise scorer, not a controlled comparison with NOTEARS or DCDI.

Summary

This chapter has isolated a modest but reusable construction. Once an interventional or perturbation law P_a is identified or directly sampled, and $P_a \ll P_0$, its Radon-Nikodym derivative transports baseline expectations to regime- a expectations. Neither identification nor absolute continuity follows from the notation. Hard continuous interventions may be singular, and finite-sample overlap can make estimated ratios unstable.

The two-density normalizing-flow plug-in is one baseline, not a privileged estimator. Future comparisons should include direct ratio estimation, inverse-probability weighting, outcome regression, Riesz methods, and doubly robust estimators with held-out diagnostics and uncertainty estimates. The experiments also show that the pairwise residual scorer is not competitive as a general causal-discovery method: favorable individual scores and small calibration gaps do not imply accurate multivariate graph recovery.

A deeper categorical account of the relationship between conditioning and intervention remains a topic for deeper study. Any such account must preserve the distinction established here: a universal construction cannot by itself identify an interventional law, establish domination, or justify a density-ratio estimator. The measure-theoretic claims in this chapter stand independently of that open categorical question.

This chapter requires considerable background reading in the foundations of probability theory.⁴⁹ In addition, we highly recommend the recent PhD thesis that provide the foundational ideas for this chapter.⁵⁰

Suggested Reading

For measure-theoretic foundations, see Billingsley [1995]. Classical causal inference is treated by Pearl [2009] and Imbens and Rubin [2015]; causal influence is quantified by Janzing et al. [2013]. van Belle [2024] develops the Kan-extension and probability background that motivated the categorical questions examined in this chapter. Mahadevan [2026e] supplies the corresponding infinitesimal semantics for intervention-generated tangent fields. The BRIDGE/SKFM framework uses density-ratio or transport fields and their Lie-bracket residuals as practical diagnostics for latent-confounded causal discovery [Mahadevan, 2026b].

⁴⁹ Patrick Billingsley. *Probability and Measure*. Wiley, 3rd edition, 1995

⁵⁰ Ruben van Belle. *Kan Extensions in Probability Theory*. PhD thesis, University of Edinburgh, 2024

Section V

Universal Decision Models

Universal Decisions with Kan Extensions

Introduction

Classical decision-making—in control, planning, and reinforcement learning—is typically framed as optimizing an objective function over actions or policies. In this chapter, we take a different perspective:

Decision-making is the problem of extending partial decisions to new contexts in a canonical way.

Category theory provides a precise formulation of this idea via *Kan extensions*. The Universal Decision Model (UDM) supplies decision objects, information fields, and admissible maps. The extension layer developed here, called a *Universal Decision Learner (UDL)*, first propagates local data to a category of probes and then assembles those probes over global contexts. The distinct indexing categories are essential for the composite to be well-typed.

With application-specific indexing categories and enrichments, this perspective provides a common semantic language for:

- generalization and interpolation (via colimits),
- constraint satisfaction (via limits),
- planning and control.

Contexts and Partial Decision Models

Let:

- $\mathcal{C}$ be a category of *contexts* (states, situations, inputs),
- $\mathcal{D} \subseteq \mathcal{C}$ be a subcategory of observed contexts,
- $J : \mathcal{D} \rightarrow \mathcal{C}$ be the inclusion functor,
- $F : \mathcal{D} \rightarrow \mathcal{E}$ be a partial decision model.

The functor F assigns to each observed context $d \in \mathcal{D}$ a decision object $F(d) \in \mathcal{E}$, where $\mathcal{E}$ is a category of decision representations (e.g., vector spaces, probability distributions, or reward semirings).

The goal is to extend F to a functor defined on all of $\mathcal{C}$.

Left Kan Extensions: Generalization by Aggregation

Definition 96 (Left Kan Extension). *The left Kan extension of F along J is a functor*

$$\text{Lan}_J F : \mathcal{C} \rightarrow \mathcal{E}$$

defined by

$$(\text{Lan}_J F)(c) = \text{colim}_{(d \rightarrow c)} F(d),$$

where the colimit is taken over the comma category $(J \downarrow c)$.

Interpretation

To make a decision at a new context c , we:

- collect all observed contexts d that map to c ,
- aggregate their decisions $F(d)$ via a colimit.

A declared left Kan extension supplies one canonical structured aggregation relative to its categories and enrichment.

Example: Interpolation

Let observed objects (0) and (1) both map to a query object (q). In an ordinary category, the resulting colimit depends on the declared arrows and codomain and is not automatically a convex average. To obtain

$$(\text{Lan}_J F)(q) = \alpha F(0) + (1 - \alpha) F(1),$$

one must work in a suitable weighted or enriched setting and declare weights $\alpha, 1 - \alpha$. Kernel regression and attention can be compared with this pattern only after their weighting and aggregation operations have been shown to realize the corresponding enriched colimit.

Right Kan Extensions: Consistency and Constraints

Definition 97 (Right Kan Extension). *The right Kan extension of F along J is a functor*

$$\text{Ran}_J F : \mathcal{C} \rightarrow \mathcal{E}$$

defined by

$$(\text{Ran}_J F)(c) = \lim_{(c \rightarrow d)} F(d),$$

where the limit is taken over the comma category $(c \downarrow J)$.

Interpretation

To assign a decision at c , we:

- consider all consequences $c \rightarrow d$,
- choose a value consistent with all known decisions $F(d)$.

A declared right Kan extension assembles an object compatible with its indexed constraint diagram.

Incorporating Rewards

To model goals, we enrich the codomain $\mathcal{E}$.

Max-Plus Semiring

A useful choice is the *max-plus semiring*:

$$(\mathbb{R} \cup \{-\infty\}, \oplus = \max, \otimes = +).$$

In this setting:

- $\oplus$ selects the best option,
- $\otimes$ accumulates rewards along transitions.

Left Kan with Rewards

In an ordered max-plus enrichment, with the path weights included in the enriched diagram, the pointwise formula can take the form

$$(\mathrm{Lan}_J F)(c) = \max_{Jd \rightarrow c} (F(d) + w(Jd \rightarrow c)).$$

where $w(d \rightarrow c)$ is the reward (or cost) associated with the transition.

Left Kan extensions enumerate and evaluate possible ways to reach c .

Right Kan with Rewards

The dual operation is the enriched limit over the comma category $(c \downarrow J)$. A numerical minimum formula follows only for an order and weight convention in which that enriched limit is a minimum; it is not a general formula for right Kan extensions.

Right Kan extensions enforce consistency with downstream constraints and goals.

Universal Decision Learners

Definition 98 (Universal Decision Learner). *Let*

$$J : \mathcal{D} \rightarrow \mathcal{P}, \quad R : \mathcal{P} \rightarrow \mathcal{C}, \quad F : \mathcal{D} \rightarrow \mathcal{E}.$$

Here $\mathcal{D}$ contains observed contexts, $\mathcal{P}$ contains rollout or consistency probes, and $\mathcal{C}$ contains global contexts. When the required extensions exist, set

$$H = \text{Lan}_J F : \mathcal{P} \rightarrow \mathcal{E}, \quad \text{UDL}_{J,R}(F) = \text{Ran}_R H = \text{Ran}_R(\text{Lan}_J F) : \mathcal{C} \rightarrow \mathcal{E}.$$

Every domain and codomain now matches: the left stage produces a functor on $\mathcal{P}$, precisely the kind of input accepted by $\text{Ran}_R : [\mathcal{P}, \mathcal{E}] \rightarrow [\mathcal{C}, \mathcal{E}]$. The formerly displayed expression $\text{Ran}_J(\text{Lan}_J F)$, with $J : \mathcal{D} \rightarrow \mathcal{C}$, is not type-correct because $\text{Lan}_J F$ has domain $\mathcal{C}$, whereas Ran_J expects an input with domain $\mathcal{D}$.

If rollout is instead formed directly on $\mathcal{C}$ using $J_0 : \mathcal{D} \rightarrow \mathcal{C}$, it must be restricted to the probe category:

$$\text{Ran}_R(R^* \text{Lan}_{J_0} F).$$

The relevant adjunctions are $\text{Lan}_J \dashv J^*$ and $R^* \dashv \text{Ran}_R$. Any closure or monad claim must live on a single functor category, for example $\text{Ran}_R R^*$ on global models; idempotence requires additional reflective or coreflective hypotheses and is not asserted here.

Interpretation

The UDL operates in two stages:

1. **Expansion (Left Kan):** generate candidate decisions and propagate rewards,
2. **Filtering (Right Kan):** enforce consistency and goal satisfaction.

Decision-making consists of expanding possibilities and filtering them to satisfy constraints.

Connection to Classical Planning

Consider a simple planning problem:

$$A \rightarrow B \rightarrow G,$$

with:

- reward at goal G : 10,
- cost per transition: -1 .

Forward Evaluation (Left Kan)

$$Q(A) = 10 - 2 = 8.$$

Backward Consistency (Right Kan)

$$V(G) = 10, \quad V(B) = 9, \quad V(A) = 8.$$

Interpretation

- Left Kan: evaluates paths,
- Right Kan: enforces consistency (Bellman-style).

Universality

Theorem 27 (Universal Property of Left Kan Extension). *For any functor $G : \mathcal{C} \rightarrow \mathcal{E}$,*

$$\text{Nat}(\text{Lan}_J F, G) \cong \text{Nat}(F, G \circ J).$$

Sketch. This follows from the universal property of colimits in the comma category $(J \downarrow \mathcal{C})$. □

Interpretation

Each Kan stage is canonical relative to its declared categories, functors, and existing (co)limits; the framework does not by itself select a numerical estimator.

Conceptual Summary

A Universal Decision Model reframes decision-making as:

- extending partial knowledge across contexts,
- aggregating possibilities via colimits,
- enforcing constraints via limits.

Kan extensions provide a semantic language for extending declared local decision data; they are not, by themselves, a theory of intelligence or a numerical learning algorithm.

Decision Making as Kan Extension

A central theme of this chapter is that decision-making can be understood as a universal construction: the extension of local information into global behavior via Kan extensions. In this section, we show that both classical planning and reinforcement learning arise naturally from this perspective.

Local Behavior and Extension

Let M be a UDM object. Associated with M is a category $\mathcal{C}_M$ of contexts (e.g., states, histories, or joint configurations), together with a subcategory $\mathcal{D}_M \subseteq \mathcal{C}_M$ representing locally observed or immediately accessible contexts. A local decision or evaluation model is given by a functor

$$F_M : \mathcal{D}_M \rightarrow \mathcal{E},$$

where $\mathcal{E}$ is a category of decision objects (e.g., real-valued rewards, value functions, or probability distributions).

The problem of decision-making is to extend F_M to all of $\mathcal{C}_M$.

Planning as Left Kan Extension

In classical planning, one evaluates candidate trajectories forward from a given context. This corresponds to aggregating all possible ways in which a context can be reached or extended.

This is captured by the left Kan extension:

$$(\text{Lan}_{J_M} F_M)(c) = \text{colim}_{(d \rightarrow c)} F_M(d),$$

where $J_M : \mathcal{D}_M \hookrightarrow \mathcal{C}_M$ is the inclusion functor.

When $\mathcal{E}$ is enriched over a max-plus semiring, this reduces to

$$(\text{Lan}_{J_M} F_M)(c) = \max_{d \rightarrow c} (F_M(d) + w(d \rightarrow c)),$$

where $w(d \rightarrow c)$ represents accumulated reward or cost.

Planning corresponds to a left Kan extension: it aggregates all forward possibilities and selects those that maximize reward.

Reinforcement Learning as Right Kan Extension

For a finite discounted MDP, the connection can be made exact rather than left as an analogy. Let

$$M = (S, \{A_s\}_{s \in S}, P, r, \gamma), \quad 0 \leq \gamma < 1,$$

with bounded rewards. Let $\mathcal{P}$ be the discrete category with one object $q_{s,a}$ for every admissible state–action pair. Let $\mathcal{C}$ contain these probe objects and one object for every state, with a morphism $s \rightarrow q_{s,a}$ for each $a \in A_s$. Write $R : \mathcal{P} \hookrightarrow \mathcal{C}$ for the inclusion.

Take $\mathcal{E}$ to be the thin category of extended real numbers in reverse numerical order, so a finite categorical limit is a numerical maximum. For a candidate value function $V : S \rightarrow \mathbb{R}$, define

$$F_V(q_{s,a}) = r(s,a) + \gamma \sum_{s' \in S} P(s' \mid s, a) V(s').$$

The comma category $(s \downarrow R)$ is discrete with one object per action in A_s . Hence the pointwise right-Kan formula gives

$$(\text{Ran}_R F_V)(s) = \lim_{(s \rightarrow Rq) \in (s \downarrow R)} F_V(q) \quad (44)$$

$$= \max_{a \in A_s} \left\{ r(s, a) + \gamma \sum_{s' \in S} P(s' \mid s, a) V(s') \right\} \quad (45)$$

$$= T(V)(s). \quad (46)$$

Thus the Bellman *backup* $V \mapsto T(V)$ is exactly the restriction of this declared pointwise right Kan extension to state objects. Bellman *optimality* is the additional fixed-point condition

$$V^* = T(V^*) = \text{Ran}_R F_{V^*}|_S.$$

Since T is a γ -contraction in the sup norm, this fixed point is unique.

This is also a literal UDL instance: take $\mathcal{D} = \mathcal{P}$, $J = \text{id}_{\mathcal{P}}$, and local data F_V . Then

$$\text{UDL}_{J,R}(F_V) = \text{Ran}_R(\text{Lan}_{\text{id}_{\mathcal{P}}} F_V) \cong \text{Ran}_R F_V.$$

For sampled MDP data, a statistical left stage may instead estimate the probe values before the right stage is applied. The universal property alone does not select that estimator or prove finite-sample convergence.

Coinduction and Right Kan Extensions

Right Kan extensions express compatibility with a diagram of outgoing structure. Fixed-point and convergence claims require additional dynamical or metric structure and do not follow merely from the existence of a limit.

In particular, the value function V in reinforcement learning is characterized as a fixed point of a Bellman operator:

$$V = T(V),$$

where T is a contraction mapping in an appropriate metric space.

This is a form of *metric coinduction*: the solution is obtained as the unique fixed point of a contractive operator.

For the discounted MDP above, the pointwise right Kan extension realizes one Bellman backup, while contraction supplies the separate fixed-point theorem.

Discounted dynamic programming iterates a Bellman backup represented by a pointwise right Kan extension in the declared probe category.

Unifying Perspective

We thus arrive at a unified view:

Framework	Kan Extension	Interpretation
Planning	$\text{Lan}_J F$	Forward aggregation of possibilities
Reinforcement Learning	$\text{Ran}_J F$	Backward consistency / fixed point

Decision-making consists of extending local information into global behavior via Kan extensions.

Bisimulation as Kan Invariance

Classical notions of bisimulation arise in the study of transition systems, where two states are considered equivalent if they cannot be distinguished by any sequence of observations or transitions. In the setting of Universal Decision Models, we generalize this notion by viewing behavioral equivalence as an invariance property under Kan extensions.

Quotient Morphisms and Abstraction

We begin by recalling that a UDM object admits quotient morphisms that collapse elements, decision spaces, and information fields. Concretely, a quotient morphism

$$q : M \twoheadrightarrow M'$$

consists of structure-preserving surjections on indices and product decision spaces. Such morphisms correspond to abstractions or coarse-grainings of the original decision system.

While quotient morphisms capture an important class of model reductions, they do not by themselves characterize behavioral equivalence. To obtain a semantic notion of equivalence, we must examine how local decision behavior extends to global behavior.

Behavioral Semantics via Kan Extensions

Let M be a UDM object. As described in Section V, we associate to M :

- a category $\mathcal{C}_M$ of contexts,
- a subcategory $\mathcal{D}_M \subseteq \mathcal{C}_M$ of locally observed contexts,
- a functor $F_M : \mathcal{D}_M \rightarrow \mathcal{E}$ encoding local decision behavior.

The global behavior of M is then given by a Kan extension:

$$\hat{F}_M := \text{Lan}_{J_M} F_M \quad \text{or} \quad \hat{F}_M := \text{Ran}_{J_M} F_M,$$

where $J_M : \mathcal{D}_M \hookrightarrow \mathcal{C}_M$ is the inclusion.

Kan Invariance

The central idea is that two models are behaviorally equivalent if their induced global behaviors are identical.

Definition 99 (Kan Invariance). *A property of UDM objects is said to be Kan-invariant if it depends only on the induced Kan extension $\hat{F}_M$, and not on the particular representation of the underlying object M .*

Definition 100 (Kan Bisimulation). *Let M and N be UDM objects with associated data $(\mathcal{D}_M, \mathcal{C}_M, F_M)$ and $(\mathcal{D}_N, \mathcal{C}_N, F_N)$.*

We say that M and N are:

- left Kan bisimilar *if there exists an equivalence of context categories under which*

$$\text{Lan}_{J_M} F_M \cong \text{Lan}_{J_N} F_N$$

naturally,

- right Kan bisimilar *if*

$$\text{Ran}_{J_M} F_M \cong \text{Ran}_{J_N} F_N$$

naturally.

Interpretation

Left Kan bisimulation identifies models that generate the same forward behavior under aggregation:

Two systems are left Kan bisimilar if they induce the same space of possible outcomes and reward accumulations.

Right Kan bisimulation identifies models that satisfy the same global consistency constraints:

Two systems are right Kan bisimilar if they admit the same coinductive value functions or fixed-point solutions.

Semantic Kernels and Quotients

Equality of one optimal value is weaker than classical MDP bisimulation. Two states can share V^* while having different rewards or transition probabilities. An MDP-bisimulation quotient is recovered only when the declared semantic functor records separating information such as the full one-step reward/transition probes or value semantics for every relevant policy. A quotient based only on V^* is merely an optimal-value quotient.

Definition 101 (Semantic kernel congruence). *Let $H : \mathcal{C} \rightarrow \mathcal{E}$ be a specified induced decision semantics. A semantic kernel congruence $\sim_H$ identifies*

precisely the object and morphism distinctions erased by H . In particular, parallel morphisms f, g satisfy $f \sim_H g$ when $H(f) = H(g)$, with the analogous object relation, and the relation is compatible with domains, codomains, identities, and composition.

Theorem 28 (Semantic kernel quotient). *Suppose the quotient by $\sim_H$ exists, and write*

$$q_H : \mathcal{C} \rightarrow \mathcal{C} / \sim_H$$

for its quotient functor. Then H factors uniquely as

$$H = \overline{H} \circ q_H.$$

If $q' : \mathcal{C} \rightarrow \mathcal{Q}$ is any other quotient preserving H , so $H = H' \circ q'$, then

$$\ker(q') \subseteq \ker(H),$$

and there is a unique functor $u : \mathcal{Q} \rightarrow \mathcal{C} / \sim_H$ satisfying

$$q_H = u \circ q'.$$

Thus the semantic kernel quotient is the coarsest quotient preserving exactly the distinctions visible to H .

Proof. Because H is constant on the object and morphism classes of $\sim_H$, the universal property of the quotient supplies the unique $\overline{H}$. If $H = H' \circ q'$, then equality under q' implies equality under H , so $\ker(q') \subseteq \ker(H)$. Hence $q'(x) \mapsto q_H(x)$, and the analogous assignment on morphisms, is well-defined and gives the unique u . $\square$

The direction of factorization is important: the semantic kernel quotient factors through every finer semantics-preserving quotient, not conversely. The result is relative to the chosen semantics H ; adding more probes refines its kernel and preserves more distinctions.

Homotopy and Observational Equivalence

The notion of Kan invariance introduced above characterizes behavioral equivalence via strict equality of the induced Kan extensions. In many settings, however, exact equality is too rigid. In particular, in causal inference and reinforcement learning, it is often more natural to consider models that are *observationally equivalent* up to a suitable notion of deformation.

This motivates a homotopical refinement of Kan invariance.

HOMOTOPY OF BEHAVIORAL FUNCTORS. Let $F, G : \mathcal{C} \rightarrow \mathcal{E}$ be functors into a category equipped with a specified homotopical structure, such as a model category or an ∞ -category. The symbol $F \simeq G$ refers to equivalence in that structure. It cannot be defined merely as an informal “continuous deformation” in an arbitrary category.

Definition 102 (Homotopy Kan Equivalence). *Two UDM objects M and N are said to be homotopy Kan equivalent if their induced Kan extensions are equivalent up to homotopy:*

$$\hat{F}_M \simeq \hat{F}_N,$$

where $\hat{F}_M$ and $\hat{F}_N$ denote either left or right Kan extensions.

INTERPRETATION. Once weak equivalences and the observable semantic functor have been declared, homotopy Kan equivalence weakens natural isomorphism to the chosen derived equivalence. Whether that equivalence preserves the observables of interest is an application-specific theorem.

A chosen observational equivalence may be represented homotopically only after the target homotopy theory and observation functor have been specified.

CONNECTION TO CAUSAL MODELS. In causal inference, distinct structural models can induce identical observational distributions. Equality of those distributions is a precise observational equivalence, but it is not automatically a homotopy equivalence. A homotopical reformulation requires an explicit category of causal models, weak equivalences, and an observation functor.

HIGHER-ORDER STRUCTURE. From this perspective, the space of UDM objects carries a higher-order structure:

- objects: decision models,
- morphisms: structure-preserving maps,
- homotopies: observational equivalences.

This suggests a possible higher-categorical research direction; no such higher category is constructed in this chapter.

OUTLOOK. Homotopical methods provide a natural framework for reasoning about invariance, abstraction, and equivalence in decision-making systems. In subsequent chapters, we will see that similar ideas arise in reinforcement learning, where value functions are defined up to contraction-based equivalence, and in causal modeling, where equivalence classes of models correspond to observational indistinguishability.

Bellman Optimality and a Conditional Sheaf Analogy

The principle of optimal substructure suggests a useful analogy with sheaf theory, but Bellman optimality does not automatically supply a Grothendieck topology, restriction functors, or a unique gluing theorem. The discussion below is therefore conditional: if an application declares a site of subproblems and a presheaf whose restriction maps preserve the relevant policy semantics, one may ask whether compatible local optima glue.

CONTEXTS AS A SITE. Let $\mathcal{C}$ be a category of contexts (e.g., states, locations, or histories) equipped with a coverage (or topology) specifying admissible decompositions of a context into subcontexts. For example, a path from San Francisco to New York that passes through Chicago can be decomposed into subpaths:

$$\text{SF} \rightarrow \text{Chicago} \quad \text{and} \quad \text{Chicago} \rightarrow \text{NY}.$$

We regard such decompositions as covers in a site structure on $\mathcal{C}$.

POLICIES AND VALUES AS SECTIONS. Associate to each context $c \in \mathcal{C}$ a set (or space) of admissible policies or value assignments. This defines a presheaf

$$\mathcal{V} : \mathcal{C}^{\text{op}} \rightarrow \mathcal{E},$$

where $\mathcal{V}(c)$ denotes the space of value functions or policies on c , and restriction maps

$$\rho_{c \rightarrow d} : \mathcal{V}(c) \rightarrow \mathcal{V}(d)$$

correspond to restricting a policy or value assignment from a larger context to a subcontext.

BELLMAN OPTIMALITY AND LOCAL OPTIMALITY. Let $V^*(c)$ denote the optimal value at context c . For decompositions that satisfy the usual optimal-substructure or segment-replacement property, the restriction of an optimal policy to a reachable continuation subproblem is optimal for that subproblem:

$$\rho_{c \rightarrow d}(V^*(c)) = V^*(d).$$

This statement is not valid for an arbitrary object called a subcontext. It requires that replacing the continuation on (d) yields an admissible global policy and preserves the relevant state and boundary conditions.

SHEAF CONDITION. This consistency requirement resembles part of a sheaf condition. To obtain an actual sheaf one must separately prove locality and unique gluing for the declared covers:

- **(Locality)** If two sections agree on all subcontexts of a cover, they agree globally.
- **(Gluing)** Given a compatible family of locally optimal sections on a cover, there exists a unique global section whose restrictions agree with the local ones.

In a particular max-plus or reward-enriched construction, a gluing operation may encode an optimal compatible combination. The sheaf axioms and uniqueness must still be proved for the declared restrictions.

Bellman optimality motivates a local-to-global analogy; a sheaf theorem requires additional declared site and gluing hypotheses.

EXAMPLE: PATHS THROUGH INTERMEDIATE STATES. Consider a path from San Francisco to New York passing through Chicago. If a global policy π^* is optimal, then its restriction to the subproblem from Chicago to New York must also be optimal. Otherwise, replacing that segment with a better policy would yield a better global policy, contradicting optimality.

Thus the suffix of an optimal path is optimal under the standard shortest-path replacement assumptions. This example motivates, but does not prove, a general gluing theorem for policies.

RELATION TO KAN EXTENSIONS. This sheaf perspective aligns with the Kan extension viewpoint. The right Kan extension

$$(\text{Ran}_J F)(c) = \lim_{(c \rightarrow d)} F(d)$$

assembles a globally compatible object from its declared diagram. In the explicit finite-MDP construction above, a pointwise right Kan extension realizes one Bellman backup. This does not by itself prove that a presheaf of policies is a sheaf.

In this sense:

- a declared pointwise right Kan extension can realize the Bellman backup,
- a separately proved sheaf condition would ensure that declared local sections restrict and glue coherently.

CONNECTION TO COINDUCTION AND DIFFUSION. The sheaf condition expresses exact compatibility. In practice, one often computes value functions iteratively, using Bellman updates or diffusion-like repair dynamics. These methods can be viewed as approximating the global section of the value sheaf by iteratively enforcing local consistency.

Under additional site and gluing hypotheses, optimal policies may be studied as compatible global sections.

CONCEPTUAL SUMMARY. We may summarize the correspondence as follows:

- restriction of policies $\leftrightarrow$ presheaf structure,
- Bellman optimality $\leftrightarrow$ a possible local-consistency law,
- value function V^* $\leftrightarrow$ a candidate global section,
- Bellman iteration $\leftrightarrow$ a possible iterative repair scheme,
- right Kan extension $\leftrightarrow$ construction of globally consistent semantics.

Conceptual Summary

We may summarize the discussion as follows:

- Quotient morphisms define *syntactic abstraction*.
- Kan extensions define *semantic behavior*.
- Bisimulation arises as *invariance under universal extension*.

Bisimulation can be represented as Kan invariance only when the declared semantic probes separate the reward and transition behavior required by the bisimulation notion.

Online Learning as a Comparator Extension

For actions $a \in A$ and losses ℓ_t , an additive enriched diagram can accumulate each fixed action's loss,

$$L_T(a) = \sum_{t=1}^T \ell_t(a).$$

The external-regret statistic compares the learner's realized loss with the minimum of this fixed-action family:

$$\text{Regret}_T = \sum_{t=1}^T \ell_t(a_t) - \min_{a \in A} L_T(a).$$

This gives a precise extension-and-comparison semantics once the history and comparator categories are declared. Sublinear regret is not implied by a Kan extension; it is an algorithm-dependent bound proved under the usual convexity, regularity, feedback, and stepsize assumptions.

Bridge to Reinforcement Learning

The next chapter shows how the same universal-construction viewpoint specializes to reinforcement learning:

- value functions arise as right Kan extensions,
- Bellman equations express coinductive consistency, and
- sufficiently rich MDP semantics can induce bisimulation-preserving quotients through their semantic kernels.

Suggested Reading

The intrinsic-model formulation of decentralized decision making begins with [Witsenhausen \[1975\]](#). The universal decision framework is developed in [Mahadevan \[2021\]](#); the categorical background comes from Kan extensions [\[Kan, 1958, Riehl, 2017\]](#) and coalgebra [\[Jacobs, 2016, Rutten, 2000\]](#).

Universal Reinforcement Learning

This chapter gives a deliberately scoped account of Universal Reinforcement Learning (URL). Coalgebras provide a general language for dynamical systems, while the probe construction below realizes the Bellman optimality *backup* for a finite discounted Markov decision process as a pointwise right Kan extension. The fixed-point and convergence results remain the classical contraction results; they do not follow from the Kan universal property alone. No new RL performance theorem is claimed.

Coalgebraic Dynamics

For an endofunctor $F : \mathcal{C} \rightarrow \mathcal{C}$, an F -coalgebra is a pair (X, α) with $\alpha : X \rightarrow F(X)$. A coalgebra morphism $h : (X, \alpha) \rightarrow (Y, \beta)$ satisfies

$$F(h) \circ \alpha = \beta \circ h.$$

Coalgebras model systems by their one-step observations and continuations [Rutten, 2000, Jacobs, 2016]. The choice of F is substantive: deterministic, probabilistic, and partially observed systems generally use different functors.

A finite stochastic MDP with a common action set may be packaged, for example, by a map

$$\alpha : S \longrightarrow (\mathcal{D}(\mathbb{R} \times S))^A,$$

where $\mathcal{D}$ is a chosen finite-distribution functor. This notation records action-indexed reward/successor laws. It does not itself specify an optimality criterion, discount factor, policy, or learning algorithm.

The Bellman Backup as a Pointwise Right Kan Extension

Let $M = (S, \{A_s\}_{s \in S}, P, r, \gamma)$ be a finite MDP with nonempty finite action sets, bounded expected rewards, and $0 \leq \gamma < 1$. Let $\mathcal{P}$ be the discrete category with one object $q_{s,a}$ for every admissible state–action pair. Let $\mathcal{C}$ contain those objects, one additional object for every state s , and a morphism $s \rightarrow q_{s,a}$ for each $a \in A_s$. Write $R : \mathcal{P} \hookrightarrow \mathcal{C}$ for the inclusion.

Take $\mathcal{E}$ to be the thin category of extended real numbers in reverse numerical order; a finite categorical limit in $\mathcal{E}$ is a numerical maximum. For a

bounded candidate value function V , define

$$F_V(q_{s,a}) = r(s,a) + \gamma \sum_{s'} P(s' \mid s,a) V(s').$$

The comma category $(s \downarrow R)$ is discrete with one object per action. The pointwise right-Kan formula gives

$$\begin{aligned} (\text{Ran}_R F_V)(s) &= \lim_{(s \rightarrow Rq) \in (s \downarrow R)} F_V(q) \\ &= \max_{a \in A_s} \left\{ r(s,a) + \gamma \sum_{s'} P(s' \mid s,a) V(s') \right\} \\ &= T(V)(s). \end{aligned}$$

Thus one Bellman backup is exactly the restriction of this declared pointwise right Kan extension to state objects. Without these probes, arrows, order convention, and the functor F_V , the slogan “value is a right Kan extension” would not be a theorem.

Theorem 29 (Discounted Bellman fixed point). *For the finite discounted MDP above, T is a γ -contraction in the sup norm. Hence it has a unique bounded fixed point V^* , and $T^n V_0 \rightarrow V^*$ for every bounded initial V_0 .*

Proof. For bounded V, W , the maximum inequality and stochasticity of P give

$$\|TV - TW\|_\infty \leq \gamma \|V - W\|_\infty.$$

The bounded functions on finite S form a complete normed space, so Banach’s fixed-point theorem applies. This is the standard discounted dynamic-programming argument [Sutton and Barto, 1998]. $\square$

Bellman *optimality* is the additional equation

$$V^* = T(V^*) = \text{Ran}_R F_{V^*}|_S.$$

The Kan calculation identifies the backup; the metric theorem supplies existence, uniqueness, and convergence.

Exact and Sample-Based Computation

For a known finite discounted MDP, exact value iteration applies T repeatedly. Exact policy iteration reaches an optimal stationary policy under the standard finite-state/action assumptions and an appropriate tie-breaking convention. Both have the same V^* , but this agreement follows from classical dynamic-programming theorems, not from naturality alone.

Monte Carlo, temporal-difference, and Q-learning methods use sampled data. Their unbiasedness, consistency, and convergence properties differ. Complete-return Monte Carlo averages are unbiased for a fixed policy under

the required sampling assumptions, whereas bootstrapped TD targets need not be unbiased. Tabular Q-learning convergence requires familiar visitation and stepsize hypotheses; nonlinear function approximation can invalidate the classical guarantee. URL therefore treats these methods as candidate computations of related Bellman semantics, not as automatically equivalent functors with identical statistical behavior.

Behavioral Equivalence and Abstraction

Probabilistic MDP bisimulation requires matching immediate rewards and matching transition probabilities after aggregation over equivalence classes. Equality of one optimal value function is strictly weaker: non-bisimilar states can have the same V^* . A semantic-kernel quotient recovers an MDP-bisimulation quotient only when the declared probes record a separating family of reward and transition observations. A kernel built from V^* alone is merely an optimal-value quotient.

MDP and PSR homomorphisms form categories once identities, composition, and the precise preservation equations are fixed. Their value-preservation properties then require the corresponding homomorphism theorems; surjectivity alone is insufficient.

Final Coalgebras and Coinduction

A final F -coalgebra, when it exists, is terminal in the category of F -coalgebras. Every F -coalgebra then has a unique morphism into it, providing canonical unfolding semantics. Lambek’s lemma states that the structure map of a final coalgebra is an isomorphism. Neither final-coalgebra existence nor a computational procedure for finding it follows for an arbitrary endofunctor.

Limits and colimits in a coalgebra category also require hypotheses. Creation or preservation results depend on the endofunctor and base category; one cannot transfer every Set construction automatically to $\text{Coalg}(F)$. For probability functors, weak-pullback preservation affects standard bisimulation results [Rutten, 2000, Sokolova, 2011].

Metric coinduction supplies reasoning principles for invariant closed sets of contractive maps [Kozen and Ruozzi, 2009]. In discounted RL, however, the direct convergence proof remains Banach contraction. Calling that computation “coinductive” is an interpretation, not a replacement for its metric hypotheses.

Information Fields and URL

Universal Decision Models add information structure: which component observes which variables or messages, and which local updates are admissible. This is essential in decentralized or asynchronous learning. A categorical di-

agram can organize those dependencies, but global convergence still requires a specified update operator and the appropriate delay, coverage, contraction, or stochastic-approximation conditions. A product diagram by itself proves neither convergence nor solvability.

A well-specified URL construction therefore has four explicit stages:

1. specify the coalgebraic dynamics;
2. declare the probes and information structure;
3. state the application-specific analytic assumptions; and
4. derive the resulting value or behavioral semantics.

Only the finite discounted Bellman construction above is established here as an exact Kan calculation; extensions to POMDPs, PSRs, non-Markovian histories, or distributed systems must be proved separately.

Summary

URL is a specification and comparison language. Coalgebras describe unfolding dynamics; pointwise Kan extensions can encode declared aggregation or compatibility operations; and metric or probabilistic analysis establishes whether an algorithm reaches the desired semantics. Keeping these roles separate prevents universal properties from being mistaken for statistical or optimization guarantees.

Suggested Reading

Coalgebra and coinduction are developed by Rutten [2000] and Jacobs [2016]; stochastic coalgebras are surveyed by Sokolova [2011]. Standard discounted-RL assumptions and convergence results appear in Sutton and Barto [1998]. Metric coinduction is treated by Kozen and Ruozzi [2009].

Deep URL with Geometric Transformers

IN THE PREVIOUS CHAPTER we formulated reinforcement learning categorically, modeling environments as coalgebras

$$\gamma : X \rightarrow T(X),$$

and representation learning as the search for approximate coalgebra morphisms. This chapter presents a neural realization of that program.

We study how Geometric Transformers (GT) and a graph-smoothness penalty compiled from a declared transition relation affect learning in one synthetic planning environment. The experiment tests structural regularization; it does not by itself establish that the learned encoder is a coalgebra morphism.

The central idea is:

Universal RL provides the structural objective. GT provides relational inductive bias. DB enforces diagrammatic coherence.

This chapter focuses on *deep universal RL*: combining coalgebraic principles with neural representation learning under explicit structural constraints.

From Coalgebra Morphisms to Deep URL

Recall that in Universal RL, learning amounts to approximating a coalgebra morphism:

$$F : (X, \gamma) \longrightarrow (Z, \hat{\gamma}),$$

such that

$$\hat{\gamma} \circ F \approx T(F) \circ \gamma.$$

In deep Universal RL (Deep URL), we parameterize:

- $F_\theta : X \rightarrow Z$ (state encoder),
- $\hat{\gamma}_\phi : Z \rightarrow T(Z)$ (latent transition predictor),

and optimize for approximate commutation under task loss.

However, unconstrained neural approximators may satisfy Bellman objectives while violating structural coherence. Deep URL therefore introduces additional diagrammatic constraints derived from declared relational structure.

Algorithm: Deep URL with GT+DB

- 1: Initialize parameters θ (encoder), ϕ (latent transition), and ψ (value/Q head).
- 2: **for** each training step **do**
- 3: Sample transition (x, a, r, x') .
- 4: Compute latent states:

$$z = F_\theta(x), \quad z' = F_\theta(x').$$

- 5: Compute Bellman loss:

$$\mathcal{L}_{\text{TD}}.$$

- 6: Compute structural loss from declared transition relation:

$$\mathcal{L}_{\text{struct}} = \sum_{(x_i, x_j) \in E} \|F_\theta(x_i) - F_\theta(x_j)\|^2.$$

- 7: (Optional) Compute higher-order diagrammatic residuals:

$$\mathcal{L}_\Delta, \mathcal{L}_\square.$$

- 8: Update parameters via

$$\mathcal{L} = \mathcal{L}_{\text{TD}} + \lambda_{\text{struct}} \mathcal{L}_{\text{struct}} + \lambda_\Delta \mathcal{L}_\Delta.$$

- 9: **end for**

Loss Decomposition as Coalgebraic Constraint

The total objective can be written as:

$$\mathcal{L} = \mathcal{L}_{\text{TD}} + \lambda_{\text{struct}} \mathcal{L}_{\text{graph}} + \lambda_{\text{diag}} \mathcal{L}_{\text{diagram}}.$$

- $\mathcal{L}_{\text{TD}}$ enforces Bellman consistency.
- $\mathcal{L}_{\text{graph}}$ enforces smoothness over the declared transition graph.
- $\mathcal{L}_{\text{diagram}}$ enforces higher-order diagrammatic coherence.

The losses restrict the parameterized hypothesis class. A graph-smoothness term alone does not enforce the coalgebra-morphism square; that stronger claim would require an explicit latent dynamics map and a measured commutator residual.

REINFORCEMENT LEARNING is naturally coalgebraic. While classical presentations emphasize value functions and Bellman equations, a structural perspective begins with the transition map itself.

An MDP can be expressed as a coalgebra:

$$\gamma : X \rightarrow T(X),$$

where:

- X is the state carrier,
- T is a transition functor packaging actions, next states, and rewards.

For an unlabelled deterministic transition one may use $(T(X)=X)$; stochastic transitions may use a distribution functor. An MDP requires a richer functor that also records actions and rewards, as in the preceding chapter.

Thus, RL can be understood as learning over a coalgebraic system.

REPRESENTATION LEARNING PROBLEM. Given $\gamma : X \rightarrow T(X)$, we seek a representation functor F and a latent transition $\hat{\gamma}$ such that the following square approximately commutes:

$$\hat{\gamma} \circ F \approx \hat{F} \circ \gamma.$$

This coalgebraic commutation principle provides a structural foundation for representation learning in RL.

Setup

We model the environment behavior as a coalgebra-like map

$$\gamma : X \rightarrow F(X),$$

where X is the state carrier and F packages transition-relevant outputs. In our synthetic line-world family:

- $X = \text{State}$ with $|X| = 32$,
- actions are left/right,
- the declared transition relation $\text{Transition} \subseteq X \times X$ defines local neighborhood structure,
- difficulty is increased via random starts, action-slip probability 0.10, and Gaussian reward noise ($\sigma = 0.05$).

MODEL FAMILIES.

- **MLP**: state-to- Q baseline without explicit transition geometry.
- **GT**: transition-aware representation learning.
- **GT+DB**: GT with an additional diagrammatic consistency penalty.

OBJECTIVE DECOMPOSITION. At the language level, training follows

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \lambda_{\text{struct}} \mathcal{L}_{\text{struct}} + \lambda_{\text{dyn}} \mathcal{L}_{\text{dyn}}.$$

For this note, $\mathcal{L}_{\text{task}}$ is a TD/Bellman loss, and GT+DB uses $\lambda_{\text{struct}} > 0$ while GT uses $\lambda_{\text{struct}} = 0$. The figure summarizes the transition geometry used to define the structural relation in our language-level specification.

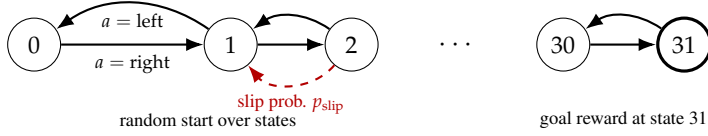


Figure 69: Synthetic line-world MDP used in experiments (hard setting): random starts, left/right actions, stochastic action slips, and noisy rewards.

Diagrammatic Backpropagation as Coalgebraic Regularization

INTERPRETATION. When rewards are noisy and transitions are stochastic, the Bellman loss alone admits many reward-compatible but geometrically inconsistent representations.

The DB term restricts learning to representations that respect the declared transition diagram. In coalgebraic terms, it penalizes deviations from commutation between observed transitions and latent transitions.

Thus GT+DB is a structurally regularized learner. Here the regularizer enforces local latent smoothness on declared edges, which is weaker than enforcing a full approximate coalgebra morphism.

The central mechanism is that a *declarative structural relation* in the language is compiled into a differentiable DB penalty.

STEP 1: RELATION DECLARATION. At the specification level, we declare a transition relation

$$\text{Transition} \subseteq X \times X.$$

In implementation terms this yields an adjacency matrix $A \in \{0, 1\}^{|X| \times |X|}$.

STEP 2: LEARNED STATE REPRESENTATION. GT produces latent state embeddings

$$H = [h_1, \dots, h_{|X|}]^T \in \mathbb{R}^{|X| \times d}.$$

STEP 3: DIAGRAMMATIC CONSISTENCY LOSS. DB enforces that adjacent states in the declared diagram carry compatible latent representations:

$$\mathcal{L}_{\text{struct}} = \frac{1}{|E|} \sum_{(i,j) \in E} \|h_i - h_j\|_2^2, \quad E = \{(i,j) : A_{ij} = 1\}.$$

Equivalent Laplacian form:

$$\mathcal{L}_{\text{struct}} = \frac{1}{|E|} \text{tr}(H^T L H), \quad L = D - A.$$

ENCODING USED IN THIS SYNTHETIC MDP. For each state s , the relation includes directed neighbors reachable by one-step actions (left/right), i.e., edges $(s, s-1)$ and $(s, s+1)$ when valid at boundaries. This yields the adjacency A used in both (i) GT message passing and (ii) DB regularization. Concretely, the same declared transition diagram supplies the edge set E in

$\mathcal{L}_{\text{struct}}$, so DB penalizes latent disagreements across one-step reachable states while allowing distant states to differ.

STEP 4: JOINT OPTIMIZATION. Training uses

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{TD}} + \lambda_{\text{struct}} \mathcal{L}_{\text{struct}}.$$

The additional term is traceable to a declared graph relation. It remains a regularization choice whose semantic validity depends on whether neighboring states should have nearby representations for the task.

INTERPRETATION. When rewards are noisy and transitions are stochastic, $\mathcal{L}_{\text{TD}}$ can admit multiple representations. The DB term penalizes edgewise latent differences; it need not remove every unstable or geometry-inconsistent solution.

Connections to PVFs and successor representations

The structural loss perspective places this work in a clear historical lineage.

PROTO-VALUE FUNCTIONS (PVFs). PVFs construct basis functions from the graph Laplacian of the MDP adjacency structure⁵¹. In that view, representation quality comes from smoothness over the transition graph.

From a geometric viewpoint:

- PVFs operate at the level of the graph Laplacian (0-form geometry).
- Successor representations capture predictive geometry over transition paths.
- GT+DB learns a transition-informed geometry jointly with task loss; it is related to, but does not subsume, the distinct PVF and successor-representation constructions.

This places GT+DB in the same lineage as spectral and predictive representations, but with an explicitly compositional and categorical formulation.

Our GT+DB formulation is closely related: the DB term is an explicit Laplacian-style smoothness penalty over learned latent states. The key distinction is that PVFs are typically computed as a *spectral preprocessing step*, while GT+DB learns representations *jointly* with task optimization.

SUCCESSOR REPRESENTATIONS (SR). Successor representations encode predictive structure of future state occupancies under a policy⁵². Like PVFs and GT+DB, SR introduces transition-informed inductive bias beyond immediate rewards.

Conceptually:

⁵¹ Sridhar Mahadevan. Proto-value functions: Developmental reinforcement learning. In *Proceedings of the 22nd International Conference on Machine Learning (ICML)*, pages 553–560. ACM, 2005. DOI: 10.1145/1102351.1102421. URL <https://doi.org/10.1145/1102351.1102421>

⁵² Peter Dayan. Improving generalization for temporal difference learning: The successor representation. *Neural Computation*, 5(4): 613–624, 1993

Model	Runs	AUC Return (95% CI)	Final Success (95% CI)	Episodes-to-Target (95% CI)	Reach Rate
GT	10	57.055 [41.767, 71.145]	0.665 [0.585, 0.725]	196.4 [160.5, 234.8]	0.900
GT+DB	10	165.078 [155.904, 175.013]	0.882 [0.868, 0.897]	55.6 [46.5, 65.3]	1.000
MLP	10	123.398 [109.843, 137.357]	0.807 [0.743, 0.863]	60.9 [28.5, 97.8]	1.000

Table 56: Ablation summary for synthetic coalgebraic MDP benchmark.

- PVF: spectral geometry from adjacency.
- SR: predictive geometry from discounted visitation.
- GT+DB: end-to-end learned geometry with explicit diagrammatic constraints from a typed specification.

WHY THIS MATTERS FOR THE PRESENT NOTE. The observed gains in this benchmark are consistent with the classical idea of using transition structure to regularize representation learning. The specification makes the source of the graph penalty explicit; broader claims require additional environments and ablations.

Results

Experiment: synthetic coalgebraic MDP ablation

The ablation contains 30 runs, ten per model family. The target criterion is rolling success at least 0.80 with window 20.

Against MLP, the point-estimate AUC delta is (41.680) and the final-success delta is (0.075). Against GT, GT+DB also has higher point-estimate AUC, final success, and faster target attainment. These results concern one synthetic environment and ten seeds per model; no general RL advantage is established.

Empirical Evidence for Structural Inductive Bias

In the synthetic line-world experiments, GT+DB has a lower mean episodes-to-target than GT. No formal hypothesis test is reported, so we do not use “significant” in the statistical sense.

From the coalgebraic viewpoint:

- The declared transition relation encodes the coalgebra structure.
- DB penalizes edgewise differences in learned representations.
- This restricts the fitted class; the run summary is consistent with improved sample efficiency in this environment.

Empirically, GT+DB reaches target success in substantially fewer episodes than GT, consistent with hypothesis restriction.

Interpretation

The principal comparison for structural learning is **GT vs GT+DB**. These models share transition-aware architecture; the difference is the DB structural term.

The empirical gap is consistent with the interpretation that DB:

- penalizes representation updates that violate declared transition geometry,
- reduces geometry-inconsistent but reward-compatible solutions under noise,
- improves sample efficiency in harder regimes.

Proposition 8 (Structural hypothesis restriction). *Let $\mathcal{H}_{GT}$ denote the hypothesis class induced by GT, and let $\mathcal{C}_{DB}$ be the set of diagrammatic constraints derived from the declared coalgebraic structure. Define*

$$\mathcal{H}_{GT+DB} = \{h \in \mathcal{H}_{GT} : h \text{ approximately satisfies } \mathcal{C}_{DB}\}.$$

Then $\mathcal{H}_{GT+DB} \subseteq \mathcal{H}_{GT}$. The inclusion is strict only if some hypothesis in $\mathcal{H}_{GT}$ violates the declared constraint at the chosen tolerance.

Remark 30. *The proposition states a structural restriction of representational capacity. Whether this improves generalization depends on the alignment between declared constraints and true environment structure.*

Remark 31. *This proposition is structural and does not by itself imply better returns. Empirical gains depend on correctness and informativeness of declared constraints.*

RELATION TO RELATIONAL MANIFOLD LEARNING. Chapter III studied higher-order simplicial constraints and reported representation metrics in its declared benchmarks.

Here, the declared transition relation induces a 1-skeleton geometry over states. DB enforces smoothness with respect to that geometry, analogous to a Laplacian regularizer.

In more complex environments with typed multi-step relations (e.g., action motifs or commuting paths), higher-order diagrammatic penalties would correspond to simplicial regularization, extending the coalgebraic view to full relational manifold learning.

Deep URL as Relational Geometry in Control

Deep Universal RL unifies three principles:

1. Coalgebraic structure (from Universal RL),
2. Relational geometry (from GT),

3. Diagrammatic coherence (from DB).

This combination yields a neural architecture that respects transition structure while optimizing reward.

In this line-world benchmark, the structural penalty is associated with better point-estimate learning metrics. Overfitting was not directly measured, and generalization to richer environments remains open.

This positions Deep URL as a structured, geometry-aware approach to reinforcement learning.

Outlook: Toward Topological Planning

The synthetic MDP experiment illustrates structural regularization at the graph level.

In richer environments containing:

- commuting action paths,
- reversible transitions,
- hierarchical subgoals,
- typed transition motifs,

higher-order diagrammatic constraints become relevant.

Extending DB+GT to incorporate 2-simplices and higher compositional structure would connect coalgebraic RL with the simplicial manifold learning framework developed in the previous chapter.

This suggests a unified program in which planning, representation learning, and relational geometry are governed by the same underlying categorical principles.

Suggested Reading

The successor representation [Dayan, 1993] and proto-value functions [Mahadevan, 2005] connect representation geometry to control. For the reinforcement-learning foundation, see Sutton and Barto [1998]; for the coalgebraic formulation used in this chapter, see Mahadevan [2025h].

Section VI

Frontiers of AGI

Consciousness

This chapter presents a categorical *research architecture* for studying the organization of conscious and unconscious processing. Its claims are deliberately limited. Coalgebras can model evolving processes; a topos can provide an internal logic; functors can describe interfaces between process families; and variational inequalities can model competition for a limited workspace. These mathematical facts do not establish that biological consciousness has this architecture, that a machine implementing it would be conscious, or that consciousness is characterized by any single categorical universal property.

Status of the proposal

Consciousness is not presently identified with an agreed computational invariant. The construction developed here is therefore a formal modeling proposal inspired by global-workspace accounts, not an empirical theory of phenomenal experience. Following the methodological separation emphasized by Marr, it is useful to distinguish a computational objective, an algorithmic architecture, and a physical realization. This chapter addresses only the first two levels. It does not infer properties of neural implementation.

Baars' global-workspace theory motivates a division between many specialized, largely parallel processes and a capacity-limited workspace in which selected contents become broadly available [Baars, 1997]. Category theory helps state an architecture with typed processes and interfaces. It does not, by itself, explain subjective experience.

Process layer: coalgebras

Let $F : \mathcal{C} \rightarrow \mathcal{C}$ be an endofunctor. An F -coalgebra is a pair (X, α) with

$$\alpha : X \longrightarrow FX.$$

A coalgebra morphism $h : (X, \alpha) \rightarrow (Y, \beta)$ satisfies

$$\beta \circ h = Fh \circ \alpha.$$

Coalgebras provide a common language for many transition systems [Jacobs, 2016, Sokolova, 2011]. For example, a labelled transition system with labels

A can be represented in **Set** using $F(X) = \mathcal{P}(A \times X)$. Probabilistic systems require a suitable distribution functor and, outside the finite case, explicit measure-theoretic hypotheses.

The existence of a final F -coalgebra is *not* automatic. It depends on the ambient category and on F . Likewise, solving a numerical fixed-point equation is not in general the same problem as constructing a final coalgebra. Any learning or asynchronous-convergence result used in this architecture must therefore state its own contraction, monotonicity, continuity, or accessibility assumptions.

Logical layer: an ambient topos

A crucial distinction is between a category of processes and an ambient category in which those processes are represented. An elementary topos has finite limits, exponentials, and a subobject classifier, and hence supports an internal higher-order intuitionistic language. A Grothendieck topos also has all small limits and colimits.

It is generally false that $\text{Coalg}(F)$ is a topos merely because F is an endofunctor. A defensible construction instead begins with a specified topos $\mathcal{E}$ —for example, a presheaf or sheaf topos—and forms coalgebra objects internally in $\mathcal{E}$. Whether the resulting category of coalgebras inherits particular limits, colimits, exponentials, or a subobject classifier requires separate preservation hypotheses on F .

Consequently, the proposed MUMBLE language should be understood as the Mitchell–Bénabou internal language of a *chosen ambient topos*, enriched with modalities only after the corresponding modal operators have been defined. The existence of such an internal language is a theorem about the topos; the claim that a brain or an AI system implements that language is an additional empirical hypothesis.

Workspace interfaces

Let $\mathcal{U}$ represent a category of typed background processes and let $\mathcal{W}$ represent workspace states. A proposed interface

$$C : \mathcal{U} \longrightarrow \mathcal{W}$$

can record which process outputs are made available to a workspace. Calling C a “consciousness functor” specifies preservation of identities and composition. It does not make consciousness a mathematical consequence of functoriality. The scientific content lies in the choice of objects, morphisms, observations, and experimentally testable constraints on C .

The reverse direction—the incorporation of workspace-mediated decisions into longer-lived processes—may be represented by a second typed interface.

Universal decision models and coalgebraic control supply candidate languages for this interaction [Mahadevan, 2021, 2025h], but their applicability depends on an explicit decision problem rather than on the consciousness interpretation alone.

Competition for limited capacity

A global workspace is capacity limited, so selection among candidate contents can be modeled as an allocation problem. The network-economy analogy assigns utilities and feasibility constraints to producers, transport channels, and workspace consumers. Under the standard continuity, convexity, and monotonicity conditions, an equilibrium may be expressed as a variational inequality [Nagurney, 1999]. Those hypotheses must be verified for a particular implementation; the analogy alone supplies neither existence nor convergence.

Asynchronous algorithms are similarly conditional. Classical convergence theorems require assumptions such as bounded communication delays, repeated component updates, and an appropriate contraction or monotone operator [Bertsekas and Tsitsiklis, 1997]. The absence of a global clock is therefore an architectural possibility, not a free convergence theorem.

What the framework does and does not establish

The proposal contributes a typed decomposition of four questions:

1. what kinds of evolving process are represented;
2. what internal logic is available in the chosen semantic universe;
3. how process outputs are transported into and out of a limited workspace;
and
4. how competing contents are selected.

Its value is organizational: each component exposes assumptions that can be changed or tested independently. It presently offers no behavioral marker that distinguishes conscious from non-conscious systems, no measurement of phenomenal experience, and no empirical validation against neural data.

Suggested Reading

Baars provides the global-workspace motivation [Baars, 1997]. Coalgebraic transition systems are developed by Jacobs [2016] and probabilistic variants by Sokolova [2011]. The full categorical proposal appears in Mahadevan [2025a]; it should be read as a formal research program subject to the limitations stated above.

Universal Imitation Games

Turing’s imitation game motivates evaluation through observable interaction. This chapter gives a categorical formulation of such tests while keeping three notions separate: categorical identity, observational equivalence under a specified probe family, and statistical indistinguishability in a finite experiment. Yoneda rigorously supports the first notion only when the full representable functor is available. Kan extensions formulate a universal extension problem, but practical preference learning is not automatically a Kan extension.

A typed imitation experiment

Turing’s original proposal replaces the ill-posed question “Can machines think?” with an operational interrogation protocol [[Turing, 1950](#)]. A modern formalization must specify at least:

- a category $\mathcal{O}$ of admissible interaction contexts;
- an outcome space or category $\mathcal{V}$;
- for each agent A , a behavior functor or stochastic kernel $B_A : \mathcal{O}^{op} \rightarrow \mathcal{V}$;
and
- a finite sampling and decision rule used by an evaluator.

The test is relative to these choices. Success on one collection of prompts does not imply equivalence on unobserved contexts, general intelligence, or consciousness.

What Yoneda licenses

For a locally small category $\mathcal{C}$, the Yoneda embedding

$$y : \mathcal{C} \longrightarrow [\mathcal{C}^{op}, \mathbf{Set}], \quad X \longmapsto \mathcal{C}(-, X)$$

is fully faithful. Hence $X \cong Y$ exactly when their full representable functors are naturally isomorphic. This is a theorem about all morphisms in the chosen category.

An imitation game observes only a restricted and usually finite probe family. If $i : \mathcal{O} \hookrightarrow \mathcal{C}$ is the inclusion of observable probes, then agreement of

$$i^*yX \quad \text{and} \quad i^*yY$$

need not imply $X \cong Y$. That implication requires the restricted probe family to be separating or dense in an appropriate sense. With stochastic responses and finite samples, even equality of restricted behavior laws is replaced by a statistical hypothesis test with stated error rates.

Thus an imitation game may be viewed as a *restricted Yoneda-style experiment*, but not as an application of Yoneda that recovers the agent’s identity or internal organization.

Static, adaptive, and population tests

The framework distinguishes three experimental regimes [Mahadevan, 2024b]:

1. a *static* game compares fixed agents under a fixed protocol;
2. an *adaptive* game changes an agent using feedback over time;
3. a *population* game studies persistence under an explicitly specified selection and reproduction process.

These are useful design categories, not a theorem that contemporary training recapitulates biological evolution. A population model requires a state space, variation operator, fitness functional, and population dynamics before evolutionary conclusions can be drawn.

Preference learning as an extension problem

Suppose $J : \mathcal{J} \rightarrow \mathcal{O}$ includes the contexts on which human comparisons have been collected, and $R_0 : \mathcal{J} \rightarrow \mathcal{D}$ records suitably typed judgments. Extending those judgments to all deployment contexts is an extension problem. If $\mathcal{D}$ has the required colimits, the left Kan extension

$$\text{Lan}_J R_0$$

is characterized by its universal property among extensions equipped with a natural comparison from R_0 .

This observation does not show that a neural reward model computes that Kan extension. One must define the categories, enrichment, comparison maps, and loss connecting the learned model to the universal construction. Practical RLHF also contains distinct stages—data collection, preference modeling, and policy optimization—and each introduces approximation and distribution shift [Christiano et al., 2023]. Human preference is moreover not identical to truth, safety, or human behavior.

Auditable claims for imitation games

A categorical imitation study should report:

1. the declared probe category and what is excluded from it;
2. the observational equivalence or discrepancy being estimated;
3. the sampling protocol, uncertainty, and evaluator reliability;
4. the extension procedure from judged to unjudged contexts; and
5. counter-tests for failures outside the original probe family.

These requirements turn the categorical language into an audit discipline. They also prevent three common conflations: behavioral similarity with object isomorphism, finite test performance with universal competence, and preference optimization with truth or alignment.

Conclusion

Turing supplies an operational test, Yoneda clarifies the strength of a *complete* relational presentation, and Kan extensions formalize one kind of local-to-global completion. Their combination gives a precise language for designing and criticizing interaction-based evaluations. It does not settle the nature of intelligence, personhood, or consciousness.

Suggested Reading

Begin with Turing’s original paper [Turing, 1950]. The universal-imitation proposal is developed in Mahadevan [2024b]. For preference-based reinforcement learning, see Christiano et al. [2023]. The categorical interpretation in this chapter is an idealized extension problem, not a claim that existing RLHF systems satisfy a Kan universal property.

Formal Verification Map

This book is accompanied by a Lean 4 formalization hosted at <https://github.com/sridharmahadevan/catagi>. The repository contains machine-checked declarations corresponding to selected categorical definitions, theorem fragments, and structural constructions developed throughout the book. The generated document `CatagiProofs.pdf` collates these declarations in one place; the purpose of this chapter is to provide the book-facing map that keeps the manuscript and the formal development synchronized.

The Lean development should be read as a *partial structural companion* to the text. It formalizes portions of the mathematical backbone of the book: categories, functors, adjunctions, simplicial and lifting-theoretic scaffolding, sheaves, coalgebras, universal decision objects, causal constructions, predictive state gluing, and coalgebraic reinforcement learning. It does *not* attempt to formalize the empirical plots, numerical tables, or implementation details of the experimental systems.

How to Use This Map

There are four levels of synchronization between the book and the Lean repository.

1. The *numbered theorem map* records the principal Lean artifacts for the book's core numbered results.
2. The *chapter-to-module map* below identifies the Lean files that formalize the main mathematical content of each chapter.
3. The *declaration-level crosswalk* records representative Lean definitions and theorems for the newer chapters added after the original formalization effort.
4. For exhaustive declaration listings, readers should consult the generated companion document `CatagiProofs.pdf` in the repository's `proofs/docs/` directory.

Numbered Theorem Map

The table below records the present synchronization between the manuscript’s main numbered results and the Lean 4 declarations in the repository. When a Lean entry cites a Mathlib declaration, the book result is formalized by reusing the standard library proof rather than reproving the fact from scratch.

Book result	Lean artifact(s)	Synchronization note
Theorem 1 / Lemma 1 (Yoneda lemma)	<code>YonedaAttention.lean</code> : <code>Mathlib yonedaEquiv</code>	Direct reuse of the standard Yoneda equivalence in Mathlib
Theorem 2 / Theorem 9 (density of representables)	<code>Diagrams.lean</code> : density-theorem specification and imported Grothendieck/Yoneda infrastructure	The full density colimit equivalence is not yet proved in the companion; finite-limit structure on causal-model categories is a different result
Theorem 10 (Causal reproducing property)	<code>ToposCausal.lean</code> : <code>causal_reproducing_property</code> , <code>causal_reproducing_full</code> , <code>causal_reproducing_faithful</code>	Realized via the fully faithful Yoneda embedding
Theorem 11 (arrow category Set ^[1] is a topos)	Standard functor-topos theorem; <code>ToposCausal.lean</code> : <code>SCM_has_limits</code> checks only a finite-limit fragment	The Lean declaration is not a certification that a constrained category of SCMs is a topos
Theorem 12 (Universal property of causal functors)	<code>CausalFunctors.lean</code> : <code>causal_functor_universal_property</code>	Stated up to isomorphism, matching the standard Kan-extension formulation
Theorem 13 (Cocontinuity of causal functors)	<code>CausalFunctors.lean</code> : <code>causal_functor_cocontinuity</code>	Formalized as the Yoneda/Kan extension of a causal functor
Theorem 14 (nonempty colimits in the stated Prob)	<code>CausalFunctors.lean</code> : <code>prob_has_colimits</code> uses an ambient Type model	The Lean surrogate does not formalize the chapter’s category of probability spaces and measure-preserving maps
Theorem 15 (Causal functors preserve colimits)	<code>CausalFunctors.lean</code> : <code>causal_functor_preserves_colimits</code>	Expressed as a <code>PreservesColimits</code> instance for the canonical Kan extension functor
Theorem 16 (Heyting algebra of subobjects)	<code>CausalFunctors.lean</code> : <code>topos_subobject_heyting</code>	Gives an explicit Heyting implication on sieves
Theorem 17 (Grothendieck/Lawvere–Tierney correspondence)	<code>JudoCalculus.lean</code> : <code>grothendieck_eq_lawvere_tierney</code>	Current Lean theorem proves the Grothendieck-to-Lawvere–Tierney direction used in the chapter
Theorem 18 (RN–Kan duality)	<code>CausalDensity.lean</code> : <code>radon_nikodym_density</code> , <code>RNkanDuality</code> , <code>RNkanDuality.ofAbsCont</code>	Measure-theoretic half comes from Mathlib; the categorical packaging is added in the repository
Former Theorem 19 (left-exact comonad coalgebras form a topos)	<code>ToposConsciousness.lean</code> : legacy identifier <code>comonad_coalgebras_topos</code>	Despite its name, this instance proves only that the coalgebra category has finite limits. It does not formalize exponentials, a subobject classifier, or a topos theorem
Kripke–Joyal fragment	<code>ToposConsciousness.lean</code> : <code>kripke_joyal_semantics</code>	Proves preservation of binary conjunction for an abstract forcing structure; it is not a formalization of full Kripke–Joyal semantics

Chapter-to-Module Map

Book chapter(s)	Lean module(s)	Representative declarations	Coverage note
Category Theory for AGI through Geometric Transformers	BasicCategory.lean, Functors.lean, Diagrams.lean, MonoidalEnriched.lean, TransformerCategory.lean, LearnCategory.lean, YonedaAttention.lean, SimplicialSets.lean, LiftingProblems.lean	categories, functors, natural transformations, Kan-style diagrammatics, enriched attention, simplices, horns, and lifting properties	Core categorical language used throughout the book, including the simplicial infrastructure later reused by the homotopy chapters
Dynamic Compositionality	DynamicCompositionality.lean, CommutatorBounds.lean	commutator energy, symmetry, nonnegativity, zero-commutator criteria	Formalizes the structural core rather than the training code
Adjoint Functors	AdjointFunctors.lean	adjunction data, unit/counit laws, universal constructions	Captures the main adjunction theorems in categorical form
Topos Causal Models	ToposCausal.lean, GrothendieckSite.lean, CausalFunctors.lean	presheaves, sieves, causal functors, subobject-classifier style structure	Covers the topos-theoretic backbone of the chapter
Judo Calculus	DoCalculus.lean, JudoCalculus.lean	SCM-style definitions, intervention structure, j -stable closure principles	Focuses on the structural rules rather than algorithm engineering
Homotopy in Language and Causal Inference; Model Categories for Causality and Language	SimplicialSets.lean, LiftingProblems.lean, TransformerCategory.lean	degenerate simplices, boundaries, horns, lifting problems, LLP/RLP, language-facing syntax and semantic scaffolds	Partial formalization of the simplicial and lifting infrastructure used by the homotopy chapters; the full localization/model-category layer remains future work
Predictive State Representations in a Topos; BASKET; ROCKET	BasketRocket.lean, PredictiveStateTopos.lean	workflows as finite posets, predictive covers, obstruction indices, gluing criteria	Newer chapter added to the synchronized formalization
Causal Density Functions	CausalDensity.lean	RN/Kan duality schema and associated categorical probability structure	Formalizes the theorem layer, not the downstream calibration experiments
Universal Decisions with Kan Extensions	Coalgebras.lean, UniversalDecision.lean	UDM objects, information fields, Witsenhausen intrinsic models, coalgebraic background reused later	Formalizes the decision-object and information-structure layer; the chapter's executable interpretation is synchronized with these definitions rather than a single omnibus theorem
Universal Reinforcement Learning	Coalgebras.lean, UniversalRL.lean	Markov chains, MDPs, Bellman operators, final coalgebra witnesses, URL information fields	Builds the RL-specific layer on top of the shared coalgebraic and UDM-style infrastructure
Deep URL with Geometric Transformers	DeepURL.lean	coalgebra residuals, structural losses, constrained hypothesis classes	Newer chapter added to the synchronized formalization
Consciousness	ToposConsciousness.lean, UniversalDecision.lean	A Heyting-algebra language structure, a conjunction rule for an abstract forcing relation, UDM objects, and a finite-limit instance for comonad coalgebras	Partial scaffolding only; it does not certify that coalgebras form a topos or that the architecture models biological consciousness

Declaration-Level Crosswalk for the Newer Chapters

The next table records the principal Lean declarations corresponding to the newer chapters and the recently expanded synchronization layer. The goal is not to reproduce every declaration from `CatagiProofs.pdf`, but to provide a precise synchronization layer from book statements to Lean names.

Book location	Book concept or statement	Lean declaration(s)	Lean file
Homotopy in Language and Causal Inference; Model Categories for Causality and Language	Simplicial primitives used to reason about paraphrase families, boundaries, and horn-filling behavior	IsDegenerate, SSet.boundary, SSet.horn	SimplicialSets.lean
Model Categories for Causality and Language	Lifting problems and left/right lifting properties underlying the model-categorical language of cofibrations and fibrations	LiftingProblem, LiftingSolution, HasLLP, HasRLP, example13_llp_specific, example14_rlp_specific	LiftingProblems.lean
Homotopy in Language and Causal Inference	Language-facing syntax and semantic scaffolds reused in the discussion of paraphrase sensitivity and semantic completion	LLMSyntaxCat, LLMSemanticCat, KNNLLMCat, KNNLLMCat.combinedDist_lambda_zero, KNNLLMCat.combinedDist_lambda_one	TransformerCategory.lean
Universal Decisions with Kan Extensions	Universal decision objects and the basic information-field layer	UDMObj, InformationField, InformationField.empty_mem, InformationField.IsSubField	UniversalDecision.lean
Universal Decisions with Kan Extensions	Witsenhausen-style intrinsic models, agent information fields, and strategy profiles	WitsenhausenModel, WitsenhausenModel.agentInfoField, WitsStrategyProfile, witsEvalCost	UniversalDecision.lean

Book location	Book concept or statement	Lean declaration(s)	Lean file
Predictive State Representations in a Topos, workflow extraction sections	Operational workflows as partially ordered plans with linearizations	<code>OperationalPlan,</code> <code>Linearization, PlanHom</code>	<code>BasketRocket.lean</code>
Predictive State Representations in a Topos, BASKET discussion	Text-to-workflow extraction interface for normalized plans	<code>BasketExtractor</code>	<code>BasketRocket.lean</code>
Predictive State Representations in a Topos, ROCKET selection discussion	Outcome-aligned workflow selection and optimal chosen plan	<code>RocketSelection,</code> <code>RocketSelection.chosen_is_optimal</code>	<code>BasketRocket.lean</code>
Predictive State Representations in a Topos, workflow normalization discussion	Normalization preserves value-complete plans	<code>RocketNormalization,</code> <code>RocketNormalization.normalized_plan_value_complete</code>	<code>BasketRocket.lean</code>
Theorem 21	Descent-style gluing of local predictive states into a global predictive section	<code>PredictiveCover,</code> <code>HasGlobalSection,</code> <code>overlapMismatch_eq_zero_of_globalSection</code>	<code>PredictiveStateTopos.lean</code>
Predictive State Representations in a Topos, mismatch definitions	A separately postulated cocycle structure and a nonnegative scalar sum of pairwise mismatch norms	<code>PredictiveObstructionClass,</code> <code>obstructionIndex,</code> <code>obstructionIndex_nonneg</code>	<code>PredictiveStateTopos.lean</code> ; the file does not prove that every computed mismatch defines a nontrivial cohomology class
Predictive State Representations in a Topos, single-context specialization	Reduction of the sheaf-theoretic formulation to the classical PSR case	<code>classical_psr_special_case</code>	<code>PredictiveStateTopos.lean</code>

Book location	Book concept or statement	Lean declaration(s)	Lean file
Universal Reinforcement Learning, foundational definitions and Bellman equations	Markov chains, MDPs, deterministic policies, and Bellman operators	MarkovChain, MDP, DeterministicPolicy, BellmanOperator, bellman_fixed_point_equation	UniversalRL.lean
Asynchronous distributed computation in URL	Componentwise updates preserve box invariants in asynchronous schemes	ComponentBox, InComponentBox, updateComponent, updateComponent_preserves_box	UniversalRL.lean
Coalgebraic URL sections	Diagrammatic URL architectures and uniqueness of maps into final witnesses	URLArchitecture, FinalCoalgebraWitness, FinalCoalgebraWitness.hom_ext	UniversalRL.lean
Information field discussion in URL and consciousness chapters	URL reuse of the UDM information-field layer	URLInformationField, URLInformationField.empty_mem	UniversalRL.lean
Deep URL with Geometric Transformers, overview section	Neural models as approximate coalgebra morphisms	DeepURLModel, coalgebraResidual, coalgebraResidual_eq_zero_of_exact	DeepURL.lean
Deep URL with Geometric Transformers, structural loss section	GT/DB structural losses and total objective decomposition	TransitionGraph, structuralLoss, structuralLoss_nonneg, totalLoss, totalLoss_nonneg	DeepURL.lean
Deep URL with Geometric Transformers, structural hypothesis discussion	Restricting the neural hypothesis class by exact structural constraints	HypothesisClass, constrainedHypothesisClass, constrainedHypothesisClass_subset, hypothesis_restriction	DeepURL.lean

Maintenance Protocol

To keep the book and repository in sync as new chapters or theorem labels are added, we follow a simple protocol.

1. Every formalized chapter should have an entry in the chapter-to-module map above.
2. Every core numbered theorem should appear in the numbered theorem map above, together with the current Lean declaration name and any scope qualifications (for example, when only the finite-limit core has been formalized).
3. Every theorem or definition in the book that is intended to be Lean checked should either carry an explicit theorem label or be listed in the declaration-level crosswalk with the corresponding Lean declaration name.
4. Whenever the formalization gains a new module, the root import file `CatagiProofs.lean`, this chapter, and the repository `README.md` should be updated together.
5. The definitive declaration inventory remains the generated `CatagiProofs.pdf`; this chapter serves as the human-readable bridge from exposition to formal proof artifact.

Suggested Reading

The Lean theorem prover is introduced by [de Moura et al. \[2015\]](#), and the shared mathematical library is documented by [The mathlib Community \[2020\]](#). For the mathematical statements formalized here, readers should pair those sources with the standard categorical references [[Mac Lane, 1971](#), [Riehl, 2017](#)].

CLIFF: An AGI Chatbot for This Textbook

CLIFF—the *Conscious Layer Interface to Functor Flow*—provides an executable companion to the theoretical program developed throughout this book. Architecturally, the conscious interface sits on top of the deeper Functor Flow causal engine. In the *Categories for AGI* setting, CLIFF helps answer questions about ideas from the textbook, course demos, and project directions, and it runs demos illustrating potential applications. It helps connect the book to runnable examples, code fragments, comparisons, and guided explorations.

It is not a separate theory layered on top of the text. Rather, it is the operational environment in which many of the book’s central constructions become inspectable on real corpora, PDF collections, and interactive queries. At the same time, CLIFF is a research prototype. Its analyses should not be treated as product endorsements, product criticism, or professional advice, and AI-based analyses can make mistakes.

The CLIFF architecture supports document retrieval, query routing, local causal model construction, causal SQL bundle generation, homotopy-based quotienting of paraphrastic claims, regime-sensitive gluing diagnostics, topic-partitioned corpus synthesis, and graphical interfaces for relational manifolds and local causal neighborhoods. The purpose of this appendix is therefore twofold:

1. to summarize these computational capabilities, and
2. to make explicit how those capabilities instantiate the categorical, homotopical, and topos-theoretic ideas developed in the book.

Why CLIFF Belongs in This Book

The central thesis of the book is that intelligence and structured reasoning are best understood through universal constructions: limits, colimits, Kan extensions, sheaf-like gluing, coalgebraic iteration, and homotopical quotients. CLIFF is the place where those claims are forced to interact with messy reality.

In practice, a user does not present a category, a functor, and a completed diagram. A user presents a query, a folder of PDFs, an article, or a vague

request such as “analyze ten recent studies of the Mediterranean diet and synthesize their joint support.” The system must then:

1. retrieve a workable local cover of evidence,
2. construct local causal representations,
3. identify which linguistic differences should be quotiented away,
4. determine where local causal views glue and where they fail to glue,
5. and present the result in a form a human can inspect.

This is exactly the setting in which the abstractions of the book become concrete. Retrieval produces a cover. Synthesis computes colimit-like aggregations. Credibility and regime checks behave like right-Kan-style consistency filters. Homotopy localization collapses paraphrastic duplication. CSQL bundles provide a structured database object over which one can test local to global compatibility. The graphical interfaces make the resulting geometry visible rather than leaving it implicit.

High-Level Architecture

At a high level, the current CLIFF stack is organized around the following loop:

1. **Query routing and acquisition.** A user request is classified into an appropriate workflow: conceptual explanation, relational manifold analysis, document synthesis, code-side investigation, or other companion tasks.
2. **Local evidence construction.** The system retrieves papers or documents, expands the local evidence set, and builds a provisional cover of the topic space.
3. **Local causal modeling.** Each document is processed by DEMOCRITUS, which extracts causal discourse, builds local causal models, and produces credibility-oriented reports.
4. **Bundle formation and quotienting.** The extracted claims are written into a CSQL bundle, where canonicalized claims, homotopy-localized views, and regime-sensitive views can be queried directly.
5. **Corpus synthesis and visualization.** CLIFF builds topic partitions, homotopy localization summaries, regime-gluing summaries, relational manifold views, and local causal model galleries so that the user can inspect both the global picture and the local neighborhoods from which it is assembled.

The important point is that the system is not a single monolithic model. It is a layered environment in which several categorical operations are exposed at different levels of the workflow.

DEMOCRITUS as the Local Causal Modeling Layer

The local modeling engine inside CLIFF is now DEMOCRITUS. Its role is to transform individual documents into structured causal artifacts. For each retrieved paper or article, DEMOCRITUS produces:

- executive summaries and credibility reports,
- extracted causal claims and relation families,
- local causal model (LCM) graphs,
- relational manifold projections of the extracted neighborhoods, and
- metadata that supports later aggregation across documents.

This local-first design is important. The book repeatedly emphasizes that one should not move immediately to a global object. Instead, one first constructs local views and only then asks whether they assemble into a coherent larger object. DEMOCRITUS is precisely the mechanism by which CLIFF obtains those local causal views from text.

The LCM graphs are especially useful in practice because they make visible the dense causal neighborhoods that would otherwise remain hidden inside prose. The relational manifold adds a second geometric perspective: it shows how the local claims cluster, while side-panel labels and hover interactions connect those clusters to interpretable topics.

Homotopy Localization and Causal Equivalence

One of the most important recent additions to CLIFF is explicit homotopy localization of causal claims. This development should be read together with the chapters on homotopy in language and causal inference and on model categories for causality and language, as well as the homotopy-Kan and j -stable discussions in Section V and in the Judo-calculus chapter.

The practical problem is simple to state. Natural language produces many paraphrases of the same causal content. If a system stores each paraphrase as an unrelated artifact, then the corpus bloats, support fragments, and the final graphs become harder to interpret. CLIFF now addresses this by normalizing claims to canonical subject–relation–object forms and then constructing homotopy-localized classes of same-direction causal variants.

The resulting diagnostics are not merely cosmetic. They measure higher-order coherence within a paraphrase family using simplicial ideas: connected components, filled triangles, open horns, and horn-fill ratios. In practical terms, this means that the system can distinguish:

- a paraphrase family that glues cleanly,
- a family that is only partially glued because some local comparisons fail to fill coherently, and

- a family that is disconnected because the supposed equivalence class is really a mixture of different downstream causal continuations.

This is exactly the bridge from the homotopy chapters to a working system. The mathematical story says that weakly equivalent linguistic realizations should be localized. The CLIFF implementation turns that principle into explicit computational artifacts visible in corpus summaries.

CSQL Bundles, Regime Gluing, and Topos-Theoretic Intuition

The current system does not stop at local causal extraction. The extracted claims are written into a CSQL bundle, which serves as a structured causal database for the run. This bundle contains raw claims, canonicalized claims, homotopy-localized views, and regime-localized views. In particular, it supports direct inspection of where claims:

- glue cleanly across multiple regimes,
- vary only by relation-family choice,
- or fail to descend because polarity or contextual interpretation changes across regimes.

This feature should be read in light of the book’s sheaf and topos themes. The issue is no longer merely whether two sentences are paraphrases. The issue is whether local causal views over different contexts or regimes can be glued into a stable global claim. In the current CLIFF interface, this appears concretely as *regime gluing*. Multi-regime glued claims are the cleanest local-to-global successes. Regime-sensitive and obstructed claims are the places where descent begins to fail.

For this reason, the CSQL bundle should be regarded as more than a storage format. It is the computational object in which local claims, covers, and gluing relations are assembled into a form that can be queried and audited. In that sense, it provides a practical bridge from extracted discourse to the book’s functorial-database and topos-theoretic viewpoints.

Topic Covers, Partitions, and Retrieval Discipline

Another important lesson from the recent evolution of CLIFF is that aggregation quality depends heavily on the quality of the retrieved cover. Broad queries such as “climate change” can wander into narrow topical basins unless the system explicitly monitors diversity and drift.

CLIFF therefore exposes topic atlases and topic partitions. These are not mere convenience features. They are practical approximations to the cover-theoretic viewpoint that runs throughout the sheaf chapters. A retrieved corpus is first decomposed into local topic neighborhoods before one attempts a flattened global synthesis. This makes it possible to see when:

- a corpus is genuinely coherent,
- a broad query has broken into several local covers,
- or retrieval has drifted into unrelated lexical impostors.

The distinction between interactive mode and deep autonomous mode is relevant here. In interactive mode, a user can curate the emerging cover before committing to a large synthesis. In deep mode, the system must rely more heavily on its own diversity guards, partitioning heuristics, and diagnostics. The topic-partition layer was added precisely because larger runs need a local decomposition before any global verdict can be trusted.

Visualization as Categorical Inspection

The current CLIFF GUI is not simply a presentation layer. It is the place where categorical structure becomes inspectable. Several views are now especially important.

Relational manifold view

The relational manifold gives a geometric picture of a document's causal or semantic neighborhood. Cluster labels, hover-sensitive topic rails, and linked topic chips allow the user to connect the geometry of the manifold to concrete topics without cluttering the embedding itself.

Local causal model galleries

The LCM gallery shows selected local causal models as rendered graphs rather than only as summaries. This is valuable because dense local neighborhoods are often more intelligible as graphs than as prose. In practice, this makes it far easier to inspect whether a document's extracted neighborhood is coherent or whether it contains obvious extraction drift.

Corpus synthesis dashboards

The corpus-level dashboard now exposes homotopy localization, normalized diagnostic support, regime gluing, topic partitions, and selected local artifacts. The point is not to force a single final conclusion too early. The point is to let the user inspect the intermediate categorical structure: local covers, quotients, partial gluings, and obstructions.

Conceptual Map from Book to System

The table below summarizes the current relationship between major theoretical themes and their CLIFF realizations.

Book concept	CLIFF realization
Left Kan extension and colimit-like aggregation	Corpus synthesis over local document evidence and candidate support surfaces
Right Kan extension and consistency	Credibility filtering, repair, regime-sensitive comparison, and coherence diagnostics
Homotopy localization	Canonical claim classes, causal equivalence families, simplicial coherence scores, and paraphrase collapse
j -stable and cover-sensitive semantics	Regime-localized claims, multi-regime gluing, and obstruction-style diagnostics
Sheaf and topos intuition	Topic covers, local-to-global assembly, and explicit visibility of descent failures
Functorial databases	CSQL bundles and queryable causal relations over normalized claim objects
Coalgebraic iteration and fixed-point behavior	Iterative deep runs, refinement loops, and stabilized summaries over evolving local evidence
Geometric/topological views	Relational manifolds, topic clusters, and graph neighborhoods supporting visual inspection

CLIFF is the computational environment in which the book's local-to-global constructions become observable on real text.

How Readers Should Use CLIFF

Readers should think of CLIFF as a research and teaching companion rather than as a black-box answer machine. Its strongest use is not simply to produce a one-line summary of a corpus. Its strongest use is to let the reader inspect:

1. what local evidence was retrieved,
2. how each document was turned into a local causal model,
3. which claims were identified up to homotopy,
4. where regime-sensitive gluing succeeds or fails,
5. and how the retrieved corpus decomposes into local topic covers.

Used in this way, CLIFF becomes an executable reading strategy for the book itself. The theoretical chapters supply the abstractions; the system shows how those abstractions behave on concrete corpora. Conversely, the behavior of the system reveals where the theory must be refined, for example by better homotopy localization, sharper cover construction, richer model-categorical quotients, or stronger topos-style descent criteria.

Repository and First Run

The current public-facing CLIFF repository is hosted at https://github.com/sridharmahadevan/CLIFF_CatAgi.git. Readers who want a lightweight first run can begin with the core repository alone, then add optional backends such as DEMOCRITUS when they want the heavier multi-document synthesis workflows.

A minimal local installation is:

```
git clone https://github.com/sridharmahadevan/CLIFF_CatAgi.git
cd CLIFF_CatAgi
python3 -m venv .venv
source .venv/bin/activate
pip install -r requirements.txt
pip install -e .
```

The core interface can then be launched with:

```
python3 -m functorflow_v3.cliff --outdir /tmp/cliff-session
```

For most readers of this book, a sensible progression is to begin with the book-guided or lighter interactive routes, and only then enable the deeper workflows that depend on additional repositories and LLM-backed stages. The repository `README.md` and the `docs/first_10_minutes.md` guide give the current route matrix and setup profiles.

Outlook

CLIFF is an executable companion to the theory in this book. The chapters on homotopy, model categories, topological causality, and universal decision models generate computational questions, while the behavior of CLIFF on large real corpora reveals where those theories need sharper operational formulations.

This reciprocal relationship is one of the main themes of the project. The book does not merely inspire the system, and the system does not merely illustrate the book. Each revises the other. In that sense, CLIFF is best understood as the executable frontier of the categorical program developed throughout *Category Theory for AGI*.

Suggested Reading

The system-facing causal components are described in the large-causal-model work [Mahadevan, 2025d], the CSQL paper [Mahadevan, 2026c], and the homotopy treatment of linguistic equivalence [Mahadevan, 2025e]. These readings explain the mathematical commitments behind the executable views presented by CLIFF.

Code Companion and Sample Repository

Alongside the Lean formalization described in the previous chapter, this book also has a separate sample-code companion hosted at <https://github.com/sridharmahadevan/Category-Theory-for-AGI-UMass-CMPSCI-692CT>. The purpose of that repository is different. The Lean repository records machine-checked mathematical structure; the sample-code repository gives the reader runnable notebooks and small experimental demos illustrating many of the book’s main constructions.

It is also useful to distinguish this repository from CLIFF. The CLIFF appendix describes a live computational system for retrieval, causal compilation, corpus synthesis, homotopy localization, and interactive inspection of real document collections. By contrast, the code companion is deliberately lighter weight. It is a pedagogical repository of notebooks, microdemos, and course-style materials meant to help readers experiment with individual constructions from the book in isolation. If CLIFF is the book’s integrated research environment, the code companion is its sandbox for small, transparent, reproducible examples.

Readers who want to experiment with the ideas in this book should treat the sample-code repository as a *computational companion*. It contains Colab notebooks, lightweight lecture materials, and a local environment file that can be used to reproduce small-scale demonstrations of categorical, topological, causal, and coalgebraic ideas from the text.

How This Appendix Differs from the Lean Map

The two companion repositories serve complementary roles.

1. The Lean repository answers the question: *which claims in the book have been formalized and machine checked?*
2. The sample-code repository answers the question: *where can I run a notebook or small experiment that illustrates the underlying idea?*

Accordingly, the code companion should not be read as a proof artifact. Its role is pedagogical and exploratory: it helps the reader inspect toy examples, run ablations, and build intuition for the algorithms and constructions developed in the main text.

Repository Layout

The repository is organized around course-style, Colab-ready materials.

Path	Purpose
README.md	Entry point listing the available notebooks, direct Colab links, and local setup instructions
notebooks/	Runnable demonstrations and small experiments corresponding to many of the book's theoretical themes
lectures/ and its README.md	Week-by-week lecture handouts and supporting course material
colab/ and its Sudoku GT/DB	Colab-oriented usage notes for one of the GT/DB examples
readme	
environment.yml	Optional local environment for readers who want to run the notebooks outside Colab
data/democritus/manifold.npz	Small bundled dataset used by the lightweight Democritus manifold example

Representative Chapter-to-Notebook Map

The map below is not exhaustive, but it provides a practical bridge from major book topics to runnable notebooks in the sample-code repository.

Book topic	Representative notebooks	Role in the book
Functorial learning and basic categorical intuition	week01_backprop_as_functor.ipynb, week04_clustering_as_functor.ipynb	Small executable demonstrations of learning procedures and clustering viewed categorically
Yoneda lemma and attention	week02_yoneda_microdemo.ipynb, week02_yoneda_self_attention_demo1.ipynb	Builds intuition for representability and the Yoneda-style interpretation of attention
Limits, colimits, pullbacks, and diagrammatic backpropagation	week03_limits_colimits_microdemo.ipynb, week03_db_colimit_energy.ipynb, week3_synthetic_av_pullback-2.ipynb	Illustrates universal constructions and diagrammatic consistency through concrete optimization problems
Geometric Transformers and causal information regimes	week01_lm_gt_vs_transformer.ipynb, week01_lm_causal_leakage_check.ipynb, week04_gt_causal_regimes_ptb_colab.ipynb, week06_gt_full_ptb.ipynb	Demonstrates architecture comparisons, causal leakage checks, and regime-sensitive language-model behavior
Kan extensions and structured completion	week05_kan_extension_transformer_demo.ipynb, week05_attention_ket.ipynb, week05_causal_attention_ket.ipynb	Gives runnable entry points for the KET/Kan-extension perspective on sequence models
Sheaves, covers, and overlap consistency	week07_sheaves_covers.ipynb, week08_topos_overlap_penalty.ipynb, week11_subobject_classifier.ipynb	Provides concrete toy models for gluing, overlap penalties, and subobject-classifier intuition
Causal discovery, j -stability, and Kan-Do reasoning	week09_causal_discovery_toy.ipynb, week10_kan_dorn.ipynb, week12_jstability_regimes.ipynb	Connects the causal chapters to executable interventions, reweighting ideas, and stable-edge experiments
Coalgebras, URL, and RL-flavored structure	week01_coalgebra_demo.ipynb, week11_coalgebra_gtdb_rl_demo.ipynb	Illustrates deterministic/stochastic coalgebras and small RL-style comparisons
Relational geometry and Democritus-style manifolds	week05_mini_democritus.ipynb, week13_democritus_manifold.ipynb, Comparing_GT_vs_UMAP.ipynb, Comparing_GT_vs_UMAP_DBLP.ipynb	Demonstrates the geometric and topological themes underlying the relational chapters
Open games and attention economy	week13_open_games_attention_economy.ipynb	A lightweight executable entry point for the attention-economy and open-games perspective

How Readers Can Use the Repository

There are two straightforward ways to use the code companion.

1. Open the notebooks directly in Google Colab using the links listed in the

repository `README.md`. This is the easiest route for readers who want to experiment without setting up a local environment.

2. Run the notebooks locally by creating the conda environment specified in `environment.yml`. This is preferable for readers who want to edit, extend, or benchmark the examples.

In either case, the intended workflow is simple: read a chapter, identify the corresponding notebook in the map above, and then run the smallest relevant example before moving to the larger experiments. This appendix therefore serves as the computational analogue of the formal-verification appendix: one points to proofs, while the other points to executable illustrations.

Suggested Reading

The notebooks are computational companions rather than primary literature. Readers seeking mathematical background should use [Spivak \[2014\]](#), [Fong and Spivak \[2018\]](#), and [Riehl \[2017\]](#); chapter-specific references appear in the Suggested Reading section preceding each notebook map.

Bibliography

- Peter Aczel. *Non-Well Founded Sets*. University of Chicago Press, 1988.
- Bernard Baars. *In the theater of consciousness: The workspace of the mind*. Oxford Univ. Press, New York, NY, 1997. URL <https://psycnet.apa.org/doi/10.1093/acprof:oso/9780195102659.001.1>.
- M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15:1373–1396, 2003.
- M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7(85):2399–2434, 2006. URL <https://jmlr.org/papers/v7/belkin06a.html>.
- Y. Bengio. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2(1):1–127, 2009.
- Y. Bengio, J. Paiement, and P. Vincent. Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, pages 177–184, 2004.
- Dimitri P. Bertsekas and John N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1997. ISBN 1886529019.
- Patrick Billingsley. *Probability and Measure*. Wiley, 3rd edition, 1995.
- Raoul Bott and Loring W. Tu. *Differential Forms in Algebraic Topology*. Springer, 1982.
- Tai-Danae Bradley, John Terilla, and Yiannis Vlassopoulos. An enriched category theory of language: from syntax to semantics, 2021. Arxiv.
- Philippe Brouillard, Sébastien Lachapelle, Alexandre Lacoste, Simon Lacoste-Julien, and Alexandre Drouin. Differentiable causal discovery from interventional data, 2020. URL <https://arxiv.org/abs/2007.01754>.

- Gunnar E. Carlsson and Facundo Mémoli. Classifying clustering schemes. *Found. Comput. Math.*, 13(2):221–252, 2013. DOI: 10.1007/s10208-012-9141-9. URL <https://doi.org/10.1007/s10208-012-9141-9>.
- Sneha Chaudhari, Varun Mithal, Gungor Polatkan, and Rohan Ramanath. An attentive survey of attention models, 2021. URL <https://arxiv.org/abs/1904.02874>.
- David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of machine learning research*, 3(Nov):507–554, 2002.
- Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences, 2023. URL <https://arxiv.org/abs/1706.03741>.
- Peter Dayan. Improving generalization for temporal difference learning: The successor representation. *Neural Computation*, 5(4):613–624, 1993.
- Leonardo de Moura, Soonho Kong, Jeremy Avigad, Floris van Doorn, and Jakob von Raumer. The Lean theorem prover (system description). In *Automated Deduction—CADE-25*, volume 9195 of *Lecture Notes in Computer Science*, pages 378–388. Springer, 2015. DOI: 10.1007/978-3-319-21401-6_26.
- Ben Dickson. How moonshot engineered a 2.8t behemoth to run in production. AlphaSignal, Sunday Deep Dive newsletter, August 2026. Production-oriented commentary on the Kimi K3 architecture.
- Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. *Discrete & Computational Geometry*, 28(4):511–533, 2002.
- Brendan Fong and David I Spivak. *Seven Sketches in Compositionality: An Invitation to Applied Category Theory*. Cambridge University Press, 2018.
- Brendan Fong, David I. Spivak, and Rémy Tuyéras. Backprop as functor: A compositional perspective on supervised learning. In *34th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2019, Vancouver, BC, Canada, June 24-27, 2019*, pages 1–13. IEEE, 2019. DOI: 10.1109/LICS.2019.8785665. URL <https://doi.org/10.1109/LICS.2019.8785665>.
- Tobias Fritz. A synthetic approach to markov kernels, conditional independence and theorems on sufficient statistics. *Advances in Mathematics*, 370:107239, August 2020. ISSN 0001-8708. DOI: 10.1016/j.aim.2020.107239. URL <http://dx.doi.org/10.1016/j.aim.2020.107239>.

Prashant Garg and Thiemo Fetzer. Causal claims in economics, 2026. URL <https://arxiv.org/abs/2501.06873>.

Robert Ghrist. *Elementary Applied Topology*. Createspace, 2014.

Mustafa Hajj, Lennart Bastian, Sarah Osentoski, Hardik Kabaria, John L. Davenport, Sheik Dawood, Balaji Cherukuri, Joseph G. Kocheemoolayil, Nastaran Shahmansouri, Adrian Lew, Theodore Papamarkou, and Tolga Birdal. Copresheaf topological neural networks: A generalized deep learning framework, 2025. URL <https://arxiv.org/abs/2505.21251>.

Christopher Hidey and Kathy McKeown. Identifying causal relations using parallel Wikipedia articles. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1424–1433. Association for Computational Linguistics, 2016.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Mark Hovey. *Model Categories*, volume 63 of *Mathematical Surveys and Monographs*. American Mathematical Society, Providence, RI, 1999. DOI: 10.1090/surv/063.

Guido W. Imbens and Donald B. Rubin. *Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction*. Cambridge University Press, USA, 2015. ISBN 0521885884.

Amin Jaber, Murat Kocaoglu, Karthikeyan Shanmugam, and Elias Bareinboim. Causal discovery from soft interventions with unknown targets: Characterization and learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9551–9561. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/6cd9313ed34ef58bad3fdd504355e72c-Paper.pdf.

Bart Jacobs. *Introduction to Coalgebra: Towards Mathematics of States and Observation*, volume 59 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2016. ISBN 9781316823187. DOI: 10.1017/CBO9781316823187. URL <https://doi.org/10.1017/CBO9781316823187>.

Bart Jacobs, Aleks Kissinger, and Fabio Zanasi. Causal inference via string diagram surgery: A diagrammatic approach to interventions and counterfactuals. *Math. Struct. Comput. Sci.*, 31(5):553–574, 2021. DOI:

10.1017/S096012952100027X. URL <https://doi.org/10.1017/S096012952100027X>.

Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning – with Applications in R*, volume 103 of *Springer Texts in Statistics*. Springer, New York, 2013. ISBN 978-1-4614-7137-0. DOI: 10.1007/978-1-4614-7138-7.

Dominik Janzing, David Balduzzi, Moritz Grosse-Wentrup, and Bernhard Schölkopf. Quantifying causal influences. *Annals of Statistics*, 41(5): 2324–2358, 2013.

Daniel Kan. Adjoint functors. *Transactions of the American Mathematical Society*, 87(2):294–329, 1958. URL <https://doi.org/10.2307/1993102>.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models, 2020. URL <https://arxiv.org/abs/1911.00172>.

Jeonghoon Kim, Byeongchan Lee, Cheonbok Park, Yeontaek Oh, Beomjun Kim, Taehwan Yoo, Seongjin Shin, Dongyoon Han, Jinwoo Shin, and Kang Min Yoo. Peri-In: Revisiting normalization layer in the transformer architecture, 2025. URL <https://arxiv.org/abs/2502.02732>.

Kimi Team. Kimi K3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026. URL <https://arxiv.org/abs/2607.24653>.

Jon Kleinberg. An impossibility theorem for clustering. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15. MIT Press, 2002.

Dexter Kozen and Nicholas Ruoizzi. Applications of metric coinduction. *Logical Methods in Computer Science*, 5(3):1–19, 2009. DOI: 10.2168/LMCS-5(3:10)2009. URL <https://lmcs.episciences.org/1168>.

Christian Léonard. A survey of the schrödinger problem and some of its connections with optimal transport. *Discrete and Continuous Dynamical Systems*, 34(4):1533–1574, 2014. DOI: 10.3934/dcds.2014.34.1533.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Prompt repetition improves non-reasoning llms, 2025. URL <https://arxiv.org/abs/2512.14982>.

- Michael L. Littman, Richard S. Sutton, and Satinder P. Singh. Predictive representations of state. In *Advances in Neural Information Processing Systems 14*, pages 1555–1561. MIT Press, 2001.
- Saunders Mac Lane. *Categories for the Working Mathematician*. Springer-Verlag, New York, 1971. Graduate Texts in Mathematics, Vol. 5.
- Saunders Mac Lane and Ieke Moerdijk. *Sheaves in Geometry and Logic a First Introduction to Topos Theory*. Springer New York, New York, NY, 1992. ISBN 9781461209270 1461209277. URL <http://link.springer.com/book/10.1007/978-1-4612-0927-0>.
- Sridhar Mahadevan. Proto-value functions: Developmental reinforcement learning. In *Proceedings of the 22nd International Conference on Machine Learning (ICML)*, pages 553–560. ACM, 2005. DOI: 10.1145/1102351.1102421. URL <https://doi.org/10.1145/1102351.1102421>.
- Sridhar Mahadevan. Universal decision models. *CoRR*, abs/2110.15431, 2021. URL <https://arxiv.org/abs/2110.15431>.
- Sridhar Mahadevan. Universal causality. *Entropy*, 25(4):574, 2023. DOI: 10.3390/E25040574. URL <https://doi.org/10.3390/e25040574>.
- Sridhar Mahadevan. GAIA: Categorical foundations of generative AI, 2024a. URL <https://arxiv.org/abs/2402.18732>.
- Sridhar Mahadevan. Universal imitation games, 2024b. URL <https://arxiv.org/abs/2405.01540>.
- Sridhar Mahadevan. Consciousness as a functor, 2025a. URL <https://arxiv.org/abs/2508.17561>.
- Sridhar Mahadevan. Decentralized causal discovery using judo calculus, 2025b. URL <https://arxiv.org/abs/2510.23942>.
- Sridhar Mahadevan. Intuitionistic j -do-calculus in topos causal models, 2025c. URL <https://arxiv.org/abs/2510.17944>.
- Sridhar Mahadevan. Large causal models from large language models, 2025d. URL <https://arxiv.org/abs/2512.07796>.
- Sridhar Mahadevan. A rose by any other name would smell as sweet: Categorical homotopy theory for large language models, 2025e. URL <https://arxiv.org/abs/2508.10018>.
- Sridhar Mahadevan. Topos theory for generative AI and LLMs, 2025f. URL <https://arxiv.org/abs/2508.08293>.

- Sridhar Mahadevan. Universal causal inference in a topos. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025g. URL <https://openreview.net/forum?id=TOhpnECT10>.
- Sridhar Mahadevan. Universal reinforcement learning in coalgebras: Asynchronous stochastic computation via conduction, 2025h. URL <https://arxiv.org/abs/2508.15128>.
- Sridhar Mahadevan. Brand-aware Democritus: Diffusion models of corporate geometry from longitudinal 10-k corpora. Working manuscript and accompanying experimental artifacts, 2026a.
- Sridhar Mahadevan. Latent confounded causal discovery via Lie bracket geometry, 2026b. URL <https://arxiv.org/abs/2606.19610>.
- Sridhar Mahadevan. Csql: Mapping documents into causal databases, 2026c. URL <https://arxiv.org/abs/2601.08109>.
- Sridhar Mahadevan. Democritus: Homotopy-localized causal discourse extraction from language. Manuscript submitted to Entropy, 2026d.
- Sridhar Mahadevan. Infinitesimal causality, 2026e. URL <https://arxiv.org/abs/2606.24621>.
- Sridhar Mahadevan. Learning in infinitesimal non-compositional sketches, 2026f. URL <https://arxiv.org/abs/2607.15107>.
- Sridhar Mahadevan. ODYSSEY: Constructing verifiable local truth-preserving foundation models, 2026g. URL <https://arxiv.org/abs/2606.27593>.
- Sridhar Mahadevan. PROMETHEUS: Automating deep causal research integrating text, data and models, 2026h. URL <https://arxiv.org/abs/2605.12835>.
- J.P. May. *Simplicial Objects in Algebraic Topology*. University of Chicago Press, 1992.
- Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2018. URL <https://arxiv.org/abs/1802.03426>.
- A. Nagurney. *Network Economics: A Variational Inequality Approach*. Kluwer Academic Press, 1999.
- Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, USA, 2nd edition, 2009. ISBN 052189560X.

- Ben Poole, Subhaneil Lahiri, Maithra Raghu, Jascha Sohl-Dickstein, and Surya Ganguli. Exponential expressivity in deep neural networks through transient chaos. *Advances in Neural Information Processing Systems*, 29, 2016. URL <https://arxiv.org/abs/1606.05340>.
- Daniel G. Quillen. *Homotopical algebra*. Springer, 1967.
- E. Riehl. *Category Theory in Context*. Aurora: Dover Modern Math Originals. Dover Publications, 2017. ISBN 9780486820804. URL <https://books.google.com/books?id=6B9MDgAAQBAJ>.
- J.J.M.M. Rutten. Universal coalgebra: A theory of systems. *Theoretical Computer Science*, 249(1):3–80, 2000. ISSN 0304-3975. DOI: 10.1016/S0304-3975(00)00056-6. URL [https://doi.org/10.1016/S0304-3975\(00\)00056-6](https://doi.org/10.1016/S0304-3975(00)00056-6). Modern Algebra.
- Karen Sachs, Omar Perez, Dana Pe’er, Douglas A. Lauffenburger, and Garry P. Nolan. Causal protein-signaling networks derived from multi-parameter single-cell data. *Science*, 308(5721):523–529, 2005. DOI: 10.1126/science.1105809.
- Samuel S. Schoenholz, Justin Gilmer, Surya Ganguli, and Jascha Sohl-Dickstein. Deep information propagation. *International Conference on Learning Representations*, 2017. URL <https://arxiv.org/abs/1611.01232>.
- Satinder P. Singh, Michael R. James, and Matthew R. Rudary. Predictive state representations: A new theory for modeling dynamical systems. In David Maxwell Chickering and Joseph Y. Halpern, editors, *UAI ’04, Proceedings of the 20th Conference in Uncertainty in Artificial Intelligence, Banff, Canada, July 7-11, 2004*, pages 512–518. AUAI Press, 2004. URL <https://dblp.org/rec/conf/uai/SinghJR04>.
- Ana Sokolova. Probabilistic systems coalgebraically: A survey. *Theoretical Computer Science*, 412(38):5095–5110, 2011. ISSN 0304-3975. DOI: <https://doi.org/10.1016/j.tcs.2011.05.008>. URL <https://www.sciencedirect.com/science/article/pii/S0304397511003902>. CMCS Tenth Anniversary Meeting.
- D. I. Spivak. Metric realization of fuzzy simplicial sets, 2009. URL https://dspivak.net/metric_realization090922.pdf.
- David I. Spivak. Functorial data migration. *Information and Computation*, 217:31–51, 2012. DOI: 10.1016/j.ic.2012.05.001.
- David I. Spivak. *Category Theory for the Sciences*. MIT Press, 2014. ISBN 9780262028134. URL <https://mitpress.mit.edu/9780262028134/category-theory-for-the-sciences/>.

Richard S. Sutton and Andrew G. Barto. *Reinforcement learning - an introduction*. Adaptive computation and machine learning. MIT Press, 1998. ISBN 978-0-262-19398-6. URL <https://mitpress.mit.edu/9780262193986/reinforcement-learning/>.

Fiona Anting Tan, Xinyu Zuo, and See-Kiong Ng. UniCausal: Unified benchmark and repository for causal text mining. In *Database and Expert Systems Applications*, volume 14148 of *Lecture Notes in Computer Science*, pages 248–262. Springer, 2023. URL <https://github.com/tanfiona/UniCausal>.

The mathlib Community. The Lean mathematical library. In *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs*, pages 367–381. Association for Computing Machinery, 2020. DOI: 10.1145/3372885.3373824.

Lewis Tunstall, Leandro von Werra, and Thomas Wolf. Companion notebooks for natural language processing with transformers. GitHub repository, 2022a. URL https://github.com/nlp-with-transformers/notebooks/blob/main/03_transformer-anatomy.ipynb. Source-code provenance for Figures 3.1 and 3.2: Chapter 3, `03_transformer-anatomy.ipynb`, BertViz attention visualization.

Lewis Tunstall, Leandro von Werra, and Thomas Wolf. *Natural Language Processing with Transformers: Building Language Applications with Hugging Face*. O’Reilly Media, 2022b. ISBN 9781098103248.

Alan M. Turing. Computing machinery and intelligence. *Mind*, 59(236): 433–460, October 1950. DOI: 10.1093/mind/LIX.236.433. URL <https://doi.org/10.1093/mind/LIX.236.433>.

Leslie G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, 1984. DOI: 10.1145/1968.1972. URL <https://doi.org/10.1145/1968.1972>.

Ruben van Belle. *Kan Extensions in Probability Theory*. PhD thesis, University of Edinburgh, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017. URL <https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html>.

- H. S. Witsenhausen. The intrinsic model for discrete stochastic control: Some open problems. In A. Bensoussan and J. L. Lions, editors, *Control Theory, Numerical Methods and Computer Systems Modelling*, pages 322–335, Berlin, Heidelberg, 1975. Springer Berlin Heidelberg.
- Tong Xiao and Jingbo Zhu. Introduction to Transformers: an NLP perspective, 2023. URL <https://arxiv.org/abs/2311.17633>.
- Ruibin Xiong, Yaru Yang, Di He, Kai Zhang, Shuxin Zheng, Hao Zheng, Chen Xing, Liangchen Liu, and Liwei Wang. On layer normalization in the transformer architecture. *arXiv preprint arXiv:2002.04745*, 2020.
- Greg Yang. Scaling limits of wide neural networks with weight sharing. *Advances in Neural Information Processing Systems*, 32, 2019. URL <https://arxiv.org/abs/1902.04760>.
- Greg Yang. Tensor programs ii: Neural tangent kernel for any architecture. *Advances in Neural Information Processing Systems*, 34, 2021. URL <https://arxiv.org/abs/2006.14548>.
- Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank J. Reddi, and Sanjiv Kumar. Are transformers universal approximators of sequence-to-sequence functions? In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL <https://openreview.net/forum?id=ByxRM0Ntvr>.
- Alessio Zanga and Fabio Stella. A survey on causal discovery: Theory and practice, 2023. URL <https://arxiv.org/abs/2305.10032>.
- Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric P Xing. Dags with no tears: Continuous optimization for structure learning. In *Advances in Neural Information Processing Systems*, 2018.

Glossary of Notation

Notation is local to a mathematical context: the same letter may denote different objects in different chapters. This glossary records the book's principal conventions rather than assigning every symbol a single global meaning. Definitions in the chapters remain authoritative.

Categories, maps, and diagrams

Notation	Meaning	Use in the book
$\mathcal{C}, \mathcal{D}$	categories	Ambient collections of typed objects and morphisms. Bold letters such as Set denote named categories.
$\text{Ob}(\mathcal{C})$	objects of $\mathcal{C}$	The object class of a category.
$\mathcal{C}(X, Y),$ $\text{Hom}_{\mathcal{C}}(X, Y)$	morphisms from X to Y	A hom-set, when the ambient category is locally small.
$f : X \rightarrow Y$	morphism with domain X and codomain Y	A typed map, computation, translation, or process.
$g \circ f$	composite of $f : X \rightarrow Y$ and $g : Y \rightarrow Z$	First apply f , then g .
id_X	identity morphism on X	The unit for composition at X .
$\mathcal{C}^{\text{op}}$	opposite category	The category obtained by reversing every morphism in $\mathcal{C}$.
$F : \mathcal{C} \rightarrow \mathcal{D}$	functor	A translation preserving identities and composition.
$\eta : F \Rightarrow G$	natural transformation	A coherent family of maps $\eta_X : F(X) \rightarrow G(X)$.
$[\mathcal{C}, \mathcal{D}]$	functor category	Functors $\mathcal{C} \rightarrow \mathcal{D}$ and natural transformations between them.
$D : J \rightarrow \mathcal{C}$	diagram of shape J	A functor used to encode a structured family of objects and maps.
ΔX	constant diagram at X	Sends every object of the indexing category to X .

Universal constructions and representation

Notation	Meaning	Use in the book
$yX = \mathcal{C}(-, X)$	Yoneda embedding of X	Represents an object by the family of maps into it.
$\widehat{\mathcal{C}} = [\mathcal{C}^{\text{op}}, \mathbf{Set}]$	presheaf category on $\mathcal{C}$	The ambient category for relational probes, contexts, and local data.
$\lim D$	limit of D	Universal compatible assembly, including products, equalizers, and pullbacks.
$\text{colim } D$	colimit of D	Universal gluing or aggregation, including coproducts, coequalizers, and pushouts.
$X \times_Z Y$	pullback	Universal object satisfying two maps into a common codomain.
$X \amalg_Z Y$	pushout	Universal gluing of X and Y along Z .
$F \dashv G$	F is left adjoint to G	A natural correspondence $\mathcal{D}(FX, Y) \cong \mathcal{C}(X, GY)$.
$\eta : \text{Id} \Rightarrow GF$	unit of an adjunction	Canonical map from an object into the right-after-left composite.
$\varepsilon : FG \Rightarrow \text{Id}$	counit of an adjunction	Canonical map from the left-after-right composite.
$\text{Lan}_K F$	left Kan extension of F along K	Universal extension characterized by maps out; often realized by colimits.
$\text{Ran}_K F$	right Kan extension of F along K	Universal extension characterized by maps in; often realized by limits.
$\int^c P(c, c)$	coend	A quotient-like universal aggregation used in attention and topological coend constructions.
$\int_c P(c, c)$	end	A universal compatibility object dual to a coend.

Geometry, topology, and local-to-global structure

Notation	Meaning	Use in the book
K, X	simplicial complex or geometric domain	Context determines whether the symbol denotes an abstract complex, a space, or data geometry.
$C_k(K)$	k -chains on K	Linear combinations of k -simplices.
$\partial_k : C_k \rightarrow C_{k-1}$	boundary map	Satisfies $\partial_{k-1}\partial_k = 0$.
$H_k(K)$	k -th homology group	Detects k -dimensional topological features.
$\mathcal{F}$	presheaf or sheaf	Assigns local data to contexts together with restriction maps.
$\mathcal{F}(U)$	sections over U	Data visible in the context or open set U .
ρ_V^U	restriction from U to $V \subseteq U$	Transports local sections to a smaller context.
$\mathbf{Sh}_j(\mathcal{C})$	category of j -sheaves	Sheaves for a Lawvere–Tierney topology j .
Ω	subobject classifier	The truth-value object of a topos.
a_j	j -sheafification	Reflects a presheaf into the category of j -sheaves.
$U \Vdash_j \varphi$	j -local forcing	The proposition φ holds at stage U under the chosen topology.
$X \simeq Y$	equivalence or weak equivalence	The precise notion is fixed by the relevant homotopical or model-categorical context.
$\text{Ho}(\mathcal{C})$	homotopy category	Localization of $\mathcal{C}$ at its weak equivalences.

Probability and causality

Notation	Meaning	Use in the book
$\mathbb{P}, P$	probability measure or distribution	The ambient observational or interventional law is stated locally.
$\mathbb{E}_P[f]$	expectation of f under P	Statistical averaging under a specified measure.
$P(Y \mid X = x)$	conditional distribution	Observation of $X = x$, not by itself an intervention.
$P(Y \mid \text{do}(X = x))$	interventional distribution	Distribution after a declared causal intervention.
$X \perp\!\!\!\perp Y \mid Z$	conditional independence	Independence of X and Y conditional on Z .
$\mathcal{G}_{\overline{X}}$	graph with incoming arrows to X removed	Graph surgery notation used in do-calculus.
$\mathcal{G}_{\underline{X}}$	graph with outgoing arrows from X removed	The complementary graph surgery operation.
DAG, CPDAG	directed acyclic graph and completed partially directed acyclic graph	Causal structures and Markov equivalence representatives.
$D_{\text{KL}}(P \parallel Q)$	Kullback–Leibler divergence	Compares two identified probability measures; it does not create a causal interpretation.
$\frac{dP}{dQ}$	Radon–Nikodym derivative	Density of P relative to Q , when it exists.
$j\text{-do}(X = x)$	cover-sensitive intervention notation	Intervention interpreted relative to the selected j -cover.

Learning, decisions, and dynamics

Notation	Meaning	Use in the book
$X \in \mathbb{R}^{n \times d}$	sequence of n vectors of width d	Standard representation of tokens or structured features.
Q, K, V	query, key, and value arrays	Transformer attention variables.
softmax	normalized exponential map	Converts attention scores or logits into normalized weights.
θ	trainable parameters	Parameters of a model, module, or policy.
$\mathcal{L}(\theta)$	scalar objective	A task-specific loss; categorical constraints may refine its construction or interpretation.
$X \xrightarrow{\gamma} F(X)$	F -coalgebra	A state-based system observed through the interface F .
νF	final F -coalgebra	Universal domain of possibly infinite behaviors when it exists.
$\pi(a \mid s)$	policy	Distribution over actions a in state s .
V^π, Q^π	state- and action-value functions	Expected return under policy π .
T^π, T^*	Bellman operators	Policy-evaluation and optimality updates.
UDM	universal decision model	A decision construction organized through universal extension.
$\text{Lan}_j F, \text{Ran}_j F$	left and right decision extensions	Universal aggregation and consistency constructions used in the decision chapters.

Boundary: How to use this glossary

When a symbol is overloaded, use its type and surrounding diagram to determine its meaning. Category theory treats typing as part of the notation: the same glyph attached to different domains, codomains, or universal properties need not denote the same construction.

Subject Index

10-K filing, 221, 243

adjunction, 207

agent, 437

agent evaluation, 441

agentic system, 243, 451

algorithm–system co-design, 123

attention

 Kan extension, 137

 Yoneda view, 43

backpropagation, 69

Basket and Rocket, 243

Bellman operator, 423

bisimulation, 423

block prediction, 177

categorical deep learning, 59

category, 23

causal claims from language, 213

causal database, 305

causal density function, 397

causal discovery, 213

causal identification, 269

causal inference

 homotopy, 343

causal masking, 127

causal synthesis, 451

causal trajectory, 221

causality

 model category, 357

CLIFF, 451

clustering

 functorial, 59

coalgebra, 23

 reinforcement learning, 423

code companion, 459

coend, 137

cofibration, 357

colimit, 49

commutative diagram

 learning signal, 69

commutator, 95

commutator energy, 191

 aggregation, 199

composition, 23

 dynamic, 95

 preservation, 31

computational realization, 459

conditioning and intervention, 409

consciousness, 437

consciousness functor, 437

coproduct, 49

count of adjunction, 207

CSQL, 305

data leakage, 127

database schema, 305

decision making, 409

Deep URL, 427

Democritus, 213

denoising, 177

depth scaling, 199

descent, 365

diagram, 49

diagrammatic backpropagation, 69

do-calculus, 269

document

 causal mapping, 305

dynamic compositionality, 95

endofunctor, 31

expected utility, 409

experimental protocol

 scaling, 199

factorization, 69

fibration, 357

formal verification, 445

formalization map, 445

foundry, 243

functor, 31

functorial database, 305

functorial learning, 31, 59

geometric deep learning, 79

Geometric Transformer, 79, 183

 depth sweep, 199

 mean-field limit, 191

 reinforcement learning, 427

gluing, 365

Grothendieck topology, 257

higher-order interaction, 95

Hodge Laplacian, 183

homotopy, 343

homotopy localization, 451

identity morphism, 23

imitation game, 441

infinite data stream, 36

infinite-width limit, 191

information flow, 127

information regime, 127

integrated information, 437

interaction, 441

intervention, 269

interventional distribution, 397

j-stability, 269

Judo calculus, 269

Kan extension

 decision, 409

 probability, 397

 Transformer, 137

Kan Extension Transformer

- agentic system, 243
- Kimi K3, 123
- Kullback–Leibler divergence, 397
- language
 - homotopy, 343
 - model category, 357
- language modeling, 177
- large causal model, 213
- Lawvere–Tierney topology, 269
- Lean, 445
- left adjoint, 207
- left Kan extension, 137
- Lie bracket
 - residual updates, 191
- limit, 49
- local predictive state, 365
- local-to-global reasoning, 257
- machine-checked proof, 445
- manifold learning, 183
- mean-field theory, 191
- mixture of experts
 - frontier model, 123
- model category, 357
- morphism, 23
- natural language
 - causal extraction, 213
- natural transformation, 31
- neural network
 - categorical model, 59
- neural tangent kernel, 191
- non-well-founded set, 23
- non-well-founded sets, 36
- notebook, 459
- observational distribution, 397
- open game, 441
- paraphrase equivalence, 343
- partial information, 177
- path consistency, 95
- path equivalence, 343
- policy, 423
- predictive state, 423
- predictive state representation, 365
- presheaf, 43
- product, 49
- proof assistant, 445
- pullback, 49
- pushout, 49
- quantization-aware training, 123
- query, 305
- Radon–Nikodym derivative, 397
- relational geometry, 183
- relational representation, 43
- representable functor, 43
- reproducibility, 459
- research interface, 451
- right adjoint, 207
- right Kan extension, 137
- sample repository, 459
- scaling laws, 199
- search space as coalgebra, 37
- self-model, 437
- sequence model, 127
- sheaf, 257
- predictive state, 365
- signal propagation, 191
- simplicial complex, 79
- simplicial message passing, 183
- state space, 23
- structural indication, 221
- structured language model, 177
- subobject classifier, 257
- temporal causal model, 221
- temporal diffusion, 221
- Tic Tac Toe, 37
- Topological Coend Transformer, 137
- topological obstruction, 79
- topos, 257
- topos causal model, 257
- Transformer, 79
- Turing test, 441
- UMAP, 183
- unit of adjunction, 207
- universal arrow, 207
- universal construction, 49
- universal decision model, 409
- universal imitation game, 441
- universal reinforcement learning, 423
 - deep, 427
- value function, 427
- weak equivalence, 357
- workflow planning, 243
- Yoneda embedding, 43
- Yoneda lemma, 43