SRIDHAR MAHADEVAN

MACHINE LEARNING FROM ENFORCING COMPOSITIONALITY

A Categorical Framework for Structural Diagnosis and Repair

VOLUME II OF THE CATEGORICAL AI RESEARCH PROGRAM

Contents

Preface

This book develops the themes first introduced in *Categories for AGI*[1] by synthesizing them around a new core principle for machine learning. Rather than beginning with a scalar-valued loss function, it begins with the failure of a declared composition. Such a failure retains information that a single number discards: what was meant to compose, how it failed, and what kind of repair might restore it.

The framework developed here is LINCS, short for Learning in Infinitesimal Non-Compositional Sketches. Its central claim is simple:

> Compositionality is a desirable structural property of learning systems, whether natural or artificial. A deviation from it is therefore a potential signal for learning, but not yet a scalar error: it is a typed, generally non-scalar obstruction recording which declared composition failed, where it failed, and along which directions. Scalar observations may be chosen to optimize against that obstruction, but they neither exhaust its content nor determine an admissible repair.

Most learning systems begin with an objective and ask how to optimize it. LINCS begins one level earlier by declaring what structure the system promises to preserve. A failure produces an obstruction that may later be observed through a norm, likelihood, test statistic, or decision functional. Those scalarizations can drive optimization, but remain accountable to the type, location, and semantics of the obstruction. Optimization is therefore placed inside an explicit structural contract rather than allowed to define it.

Design principle

Declare the structural promise before selecting the scalar diagnostic. The same residual magnitude can represent different failures and therefore authorize different repairs.

Why category theory?

Modern machine learning already draws on a powerful mathematical toolkit. Calculus describes local variation and supplies gradients.

[1] Mahadevan [2026c], revised July 25, 2026.

Graph theory represents incidence and dependence. Probability and statistics quantify uncertainty, evidence, and generalization. Optimization searches for parameters or policies that improve an objective. LINCS uses all of these. Category theory is needed for a different question: *what is the structure that these operations are supposed to preserve when they are composed?*

Suppose a learning system is represented by a diagram

$$D : J \longrightarrow \mathcal{C}.$$

The indexing category J records typed components and their legal composites; the target category $\mathcal{C}$ records the objects and maps that realize them. A sketch can additionally declare that two paths agree, that a cone is limiting, that a cocone is colimiting, that a section descends, or that a construction is functorial. These are not merely numerical properties of isolated parameters. They are obligations among operations. When an obligation fails, the failed diagram supplies the typed obstruction from which LINCS begins.

Set theory can encode all of these constructions, and category theory need not be advertised as a stronger foundation in an absolute sense. Its advantage is one of *native expression*. Set theory foregrounds membership and elements. Category theory foregrounds maps, composition, universal properties, and transport between representations [Mac Lane, 1998]. Lawvere's functorial semantics makes the same point for theories: generators and equations can be specified abstractly, while their models are structure-preserving realizations in a chosen category [Lawvere, 1963]. This separation between a declaration and its realizations is precisely what a learning sketch requires.

This choice also makes the framework representation-aware without making it presentation-bound. Functors compare realizations; natural transformations compare structure-preserving updates; limits and colimits express assembly; adjunctions express canonical transport; sheaves separate local consistency from global realization; and tangent structure imports calculus without reducing the declaration to coordinates. Category theory therefore does not replace the familiar mathematics of machine learning. It provides the organizing language in which their outputs can be composed, compared, and audited.

Design principle

Use category theory to state the invariant structural promise; use graphs, calculus, probability, statistics, and optimization to represent, observe, test, and repair particular realizations of that promise.

Language	Native question	Role inside LINCS
Set theory	What are the elements and subsets?	Represents underlying data, states, and parameter spaces
Graph theory	Which vertices or variables are incident or dependent?	Represents wiring, neighborhoods, and sparse interaction structure
Calculus	How does a quantity change locally?	Constructs tangent probes, sensitivities, and update directions
Probability and statistics	What is uncertain, supported, or identifiable?	Quantifies evidence and supplies statistical admission tests
Optimization	Which available candidate best improves an objective?	Searches within a declared repair language
Category theory	What must compose, transport, or glue, and in what type?	Declares the structural contract and types its failure

Table 1: The mathematical languages are complementary. Category theory supplies the compositional declaration within which the others operate.

From causal models to models of learning

There is a second route into LINCS, through the causal revolution in statistics and artificial intelligence. Pearl's critique of mainstream machine learning is not simply that its predictions can be inaccurate. It is that systems trained primarily on associations often lack an explicit model of the mechanisms that produced those associations, and therefore lack a principled object on which to perform an intervention [Pearl, 2009b, 2018, Pearl and Mackenzie, 2018]. A predictor can estimate $P(Y \mid X = x)$ without representing what it would mean to replace the mechanism for X, hold the other mechanisms invariant, and ask for $P(Y \mid \mathrm{do}(X = x))$.

McCarthy posed an antecedent challenge in terms of *appearance and reality* [McCarthy, 2007]. Sense data provide partial, view-dependent appearances; intelligence must reason about the comparatively stable objects and mechanisms that generate them. Recovering reality from appearance is an inverse problem and is generally non-unique. More fundamentally, the vocabulary of the explanation need not already occur in the observations. Atomic theory required theoretical objects that were not present as patterns in unaided sense data. Democritus proposed the ontological idea centuries before Dalton made atoms and molecules elements of a quantitatively constrained scientific theory.

McCarthy and Pearl therefore expose complementary limitations. Classification of appearances does not by itself construct a model of

their hidden reality; conditioning on observations does not by itself determine how that reality would respond to action. A learning framework adequate to both challenges must permit theoretical objects behind appearances and controlled changes to the mechanisms connecting them.

Causal inference addresses this limitation by making a domain model explicit. Structural causal models, structural equation models, potential outcomes, and related formalisms specify enough structure to distinguish observation from action. Their variables and mechanisms are normally grounded in a domain such as epidemiology, biology, economics, or policy. Once that structure has been declared, an intervention can be represented as a controlled change to part of the model, followed by propagation through what remains invariant.

LINCS imports this methodological lesson into machine learning at a different level. Its primary model need not be a causal model of a biological or economic system. It is a *model of the learning system itself*: its objects, transformations, data flows, compositional obligations, observers, allowed edits, and admission tests. A learning sketch makes that system inspectable and actionable. Non-compositionality localizes a structural discrepancy; the repair language specifies what may be changed; and admission tests whether the proposed change preserves the promises that were not targeted.

Causal inference	LINCS analogue	Shared discipline
Causal model	Learning sketch and its realization	Make the relevant structure explicit before acting
Structural mechanism	Declared component, path, or universal construction	Give a proposed change a typed target
Intervention target	Localized obstruction and registered repair site	Separate diagnosis from the authority to change
Mechanism replacement	Typed repair or sketch edit	Change the target while preserving declared non-target structure
Post-intervention law	Behavior of the repaired realization	Propagate the change through the model
Experimental validation	Admission on independent probes	Do not identify proposal with successful intervention

Table 2: LINCS inherits the model-and-intervention discipline of causal inference while applying it to learning systems more generally.

The comparison also raises a proof-theoretic question. Causal models provide the semantics of intervention, while do-calculus supplies rules for transforming observational and interventional expressions under explicit graphical conditions. Chapter 1 develops the corresponding

distinction for LINCS. Its axioms specify admissible structural-learning models. It begins with a Universal Obstruction Axiom: every admissible diagnostic factors uniquely through a typed obstruction object, whose triviality is equivalent to satisfaction of the declared compositional obligations. Additional structural laws govern tangent lift, descent, and repair; a candidate repair calculus then gives rules for observer substitution, presentation elimination, localization and gluing, tangent transport, target surgery, and conservative theory extension. These rules preserve declared structural judgments. They do not identify causal effects unless the learning sketch also carries the required causal semantics.

The analogy supports several scales of action. A *parameter intervention* moves within a fixed realization. A *component intervention* replaces a morphism, module, policy, adapter, warrant, or skill. An *architectural intervention* changes routes, interfaces, covers, or composition laws. A *declaration intervention* augments the sketch with new objects, arrows, probes, or axioms. Ordinary optimization usually occupies the first level. The later levels require an explicit language of structural change and stronger evidence that the unaffected obligations remain intact.

This does not make every LINCS repair causal in Pearl's semantic sense. A repair becomes a causal intervention only when the relevant arrows denote mechanisms and the edit is connected to a realizable intervention protocol. Otherwise it is a more general *structural intervention*. Category theory supplies the typing, composition, and invariance conditions; domain knowledge and experimental design supply causal meaning. LINCS is therefore best understood as extending the discipline of causal modeling, not as declaring all learning to be causality.

Design principle

Make the learning system itself an actionable model. A proposed update should state what structural component it changes, what it holds invariant, how the change propagates, and which independent observations would justify its admission.

The observability caveat

The central claim contains a meta-level assumption that should be made explicit: a failure of compositionality must be observable through some available diagnostic. It is not enough for a declared diagram to fail in the mathematical model. The learning system, its designer, or an accompanying audit process must be able to obtain a witness that distinguishes the failure from ordinary variation.

Let $O(D)$ denote the typed obstruction associated with a realized

diagram D. In practice the learner does not inspect $O(D)$ directly. It sees an observation profile

$$\mathrm{Obs}_\Lambda(D) = \big(\omega_\lambda(O(D))\big)_{\lambda\in\Lambda},$$

where the maps ω_λ may be norms, tests, probes, comparison maps, counterfactual outcomes, human judgments, or instrument readings. After quotienting decision-null variation, the observer family should separate the obstruction classes on which the system is authorized to act:

$$[O_1] \neq [O_2] \quad\Longrightarrow\quad \omega_\lambda(O_1) \neq \omega_\lambda(O_2) \quad \text{for some } \lambda \in \Lambda.$$

This is an *observability condition*, not a theorem supplied by LINCS. If every registered observer vanishes on a genuine actionable obstruction, the system has a structural blind spot. No repair rule driven only by those observations can be expected to correct it.

Observability is always relative to a declaration, a probe family, and a quotient. A diagnostic may detect that something failed while remaining unable to identify which typed obligation failed or what caused it.

Reinforcement learning makes the distinction concrete. An agent playing backgammon may know the legal moves and eventually observe whether it won or lost. That terminal signal does not by itself reveal the compositional structure of a Bellman equation, much less localize a failed Bellman factorization. Bellman observability requires additional representational commitments: states or predictive histories, rewards, transition structure, and comparisons between one-step and continued value. The game supplies an outcome; the learning architecture supplies the structural observer.

Natural cognition raises the deeper question of where such observers and declarations originate. The debate surrounding universal grammar asks how much language-specific structure is available prior to, or constrains, language learning [Chomsky, 1965]. This should not be identified uncritically with semantic compositionality, but it illustrates the general problem: a learner cannot diagnose every structural regularity unless some representational resources make that regularity expressible.

Developmental studies offer a related example. Spelke's core knowledge program argues that human cognition is founded in part on early-developing, domain-specific representational systems [Spelke, 2000]. A later synthesis identifies systems for objects, actions, number, and space, with a possible further system for social partners [Spelke and Kinzler, 2007]. On this account, infants do not begin with an undifferentiated stream and learn every object and relation from scratch. Early structural resources constrain what can be tracked, compared, and combined. Whether a particular resource is innate, learned, culturally scaffolded, or engineered is an empirical question; LINCS requires only that the operative structural promise and its observers be declared.

There are consequently four possible sources of observability:

1. the environment supplies an explicit witness, such as a violated rule or failed test;
2. a system designer registers structural probes and audit maps;
3. a learner acquires observers from repeated intervention and repair;
4. an organism or architecture begins with structural priors that make certain failures salient.

The third possibility opens an important outer loop. A LINCS system can learn not only how to repair under a fixed observer family, but also where its current observers fail to predict admission outcomes. It may then propose a new probe, refine its cover, or enlarge its sketch. The new observer must itself be validated on held-out failures; otherwise the system merely changes what it is able to notice in order to certify its preferred repair.

The dual issue is *controllability*. Even if the probes reveal and localize an obstruction, the registered repair language may contain no admissible path from the current realization to one satisfying the declaration. A system can therefore be observable but uncontrollable: it can diagnose a failure yet must abstain from repairing it. Conversely, a system with many available interventions but no observer capable of discriminating their relevant effects is controllable only in a dangerous, unlicensed sense. Viewed as a control system, LINCS makes both assumptions relative and typed: the probe family determines which obstruction classes are observable, while the repair language determines which admissible classes are reachable. Admission then asks whether the reached realization is supported, stable, and semantically licensed.

Boundary

LINCS assumes neither perfect observability nor an innate universal compositional grammar. It assumes that each claimed repair is supported by a declared observer family capable of detecting the relevant obstruction and by a repair language capable of reaching the proposed realization. The completeness of the observers and the reachability of the repair are themselves auditable and revisable modeling claims.

Compositionality is not compression

An influential account of learning treats it as the discovery of a short description: a good model compresses the data by exposing regularity, and model selection balances the length of the model against the length of what remains unexplained.[2] LINCS neither rejects this account nor

[2] This viewpoint includes minimum description length [Rissanen, 1978, Grünwald, 2007] and its foundations in algorithmic information theory [Li and Vitányi, 2019].

takes it as the definition of learning. Description length is a scalar quantity relative to a coding language. A compositionality obstruction is a typed failure relative to a structural declaration. They answer different questions.

A compact model can encode a systematic violation of an essential composition, while a structurally faithful model may require extra description to preserve sources, qualifiers, uncertainty, or compatibility across contexts. Conversely, reusable compositional structure often makes compression possible. The relationship is therefore productive but not an equivalence. Compression may rank admissible models or repairs, regularize a search, or become one coordinate of an observation profile; it cannot by itself say which structural promise was broken or whether a shorter repair is semantically licensed.

A sharper limitation appears when discovery requires changing the language in which descriptions are compared. Zahavy's position paper *LLMs Can't Jump* uses Einstein's path to general relativity to argue that compression of observations and deduction from established axioms do not by themselves explain the abductive step that proposes new axioms [Zahavy, 2026]. Whatever one concludes about the paper's stronger claim concerning present LLMs, it identifies an important boundary: selecting a concise model inside a fixed representational frame is different from inventing and validating a new frame.

LINCS makes that boundary expressible. Repair may change a realization inside a declared sketch, or it may propose a conservative change to the declaration itself. The latter can add objects, paths, probes, quotients, or axioms and then ask whether old evidence transports and newly required compositions hold. This does not prove that LINCS can generate a creative jump. It supplies a typed language in which such a jump can be proposed, compared with the old frame, and admitted or rejected without confusing novelty with compression.

The recurring workflow

Across the theory and all ten application systems, the same six-stage workflow provides the organizing spine of the book.

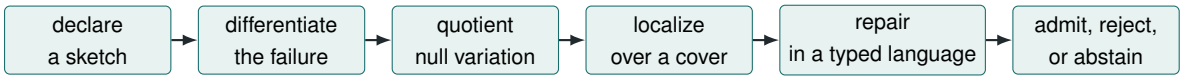


Figure 1: The recurring LINCS workflow. A run may stop before admission when the obstruction is unidentifiable, the cover is incomplete, the repair is unsupported, or validation fails.

Declaration fixes the objects, arrows, equations, and universal properties that give success a type. *Differentiation* turns a failed composite into a local sensitivity or interaction signature. *Quotienting* removes variation that is null for the declared decision. *Localization* assigns the surviving obstruction to a cover of modules, agents, contexts, skills, re-

lations, or argument fragments. *Repair* searches only within a registered language of admissible changes. *Admission* then tests the candidate against evidence independent of the signal that proposed it. Rejection and abstention are therefore successful outcomes of a well-specified workflow, not merely failures to optimize.

This separation is the practical heart of LINCS. A residual can diagnose a broken promise without identifying its cause. A localized cause need not support a feasible repair. A repair that reduces the training obstruction need not preserve meaning. Each transition requires its own assumptions and its own certificate.

What the book develops

PART I ESTABLISHES THE CATEGORICAL FOUNDATIONS. Chapter 0 supplies a working language of categories, functors, natural transformations, enriched categories and sketches, universal constructions, sheaves, and tangent structure. Chapter 1 then introduces the Universal Obstruction Axiom, states its accompanying structural laws, and develops a typed repair calculus. Chapter 2 introduces learning by repair and the six-stage workflow. Chapters 3 and 4 develop learning sketches, factorization obstructions, tangent lifts, and infinitesimal non-compositionality. Chapter 5 specializes these ideas to intervention protocols and infinitesimal causality. Chapter 6 constructs infinitesimal categorical databases, separating pointwise tangent instances, tangent-stable database sketches, and bracket enrichment. Chapter 7 develops Infinitesimal Decisions as the tangent layer of Universal Decision Learning. Chapter 8 gives the deep computational realization, Deep Learning in Non-Compositional Sketches (DLINCS), while Chapter 9 makes quotienting, localization, repair, and admission explicit as separate mathematical and engineering obligations. Chapter 10 then assembles these pieces into CoLT, a general architecture for reasoning through typed diagnosis, repair, and admission.

PART II TURNS THE FRAMEWORK INTO LEARNING SYSTEMS. BRIDGE and SKFM use Lie-bracket geometry to screen for latent-confounded causal structure. LINCS–KET treats attention and diffusion architectures through Kan-extension structure. ALLORA studies composable low-rank neural adapters. LASKO formulates skill optimization over Lie algebroids. GIRL recasts reinforcement learning as infinitesimal, structurally constrained repair. These chapters deliberately span different objects of learning: graphs, architectures, adapters, skills, and policies.

Part III focuses on relational learning and reasoning. LINCS–RLHF replaces a single scalar reward with typed preference obstructions and guarded admission. RADAR repairs relational manifold structure rather than isolated embeddings. SID uses sheaf descent to coordinate local agents while preserving the boundary between local evidence and global claims. LINCS–Toulmin applies the same grammar to argumentation, where claims, grounds, warrants, backing, rebuttals, and qualifiers must glue without erasing provenance or disagreement. Odyssey and Prometheus then extend the framework to trustworthy foundation-model foundries, separating artifact construction from structural validation, argument scrutiny, and durable admission.

Part IV extracts the common design. Chapter 21 presents a pattern language for new lincs systems. Chapter 22 develops the research frontier, from learned covers and higher interactions to homotopical repair, coalgebraic stabilization, statistical tangent sites, and mechanized verification.

The applications are not instances of one universal loss. They share a discipline: state the promise, retain the failed diagram, separate signal from semantics, make the learning system itself an actionable model, and require a certificate before changing it.

How to read the book

New readers can begin with Chapters 0–2. The theoretical route continues through Chapters 3–10 and 21. The causal route connects Chapters 5–6 with Chapter 11; Chapter 7 gives the general tangent semantics of decision making, while Chapters 12–14 treat architectures, adapters, and skills; and Chapters 15–20 treat trajectories, preferences, relations, distributed claims, arguments, and trustworthy foundation-model foundries. This is an advanced companion to *Categories for AGI*; readers seeking a slower introduction may begin with that book.

The boundary of the claim

LINCS does not assert that every compositional failure has a unique cause, that every obstruction supports a repair, or that categorical structure eliminates uncertainty. It makes those boundaries explicit by distinguishing diagnosis from identification, intervention, repair, and admission.

Nor does it assert that every repair is a causal intervention. The broader claim is methodological: mainstream learning becomes more inspectable when its own architecture and obligations are modeled explicitly enough to support localized, invariant-preserving action. Causal semantics is an additional achievement when those modeled components are grounded as mechanisms.

That distinction matters as learning systems become modular, agentic, and distributed. A global score can improve while a local contract is violated; successful components can fail when composed; and a plausible repair can destroy the semantics used to justify it. This book treats such cases as central objects of learning.

Boundary

A LINCS certificate supports only the claim registered by its declaration, cover, probe family, quotient, and admission test. It does not license a stronger causal, semantic, or safety conclusion without additional evidence.

The durable question running through the book is therefore not simply whether a model can reduce a loss. It is:

> Which structural promise failed, how does that failure move, and what part of the modeled learning system may be changed while the rest is held fixed? What independent evidence licenses that repair without changing the promise merely in order to pass?

The chapters that follow develop a language, workflow, and family of systems for answering that question.

A Trail Map of LINCS

The Preface explains why compositional failure can become a learning signal. This trail map shows how the pieces fit together before the book develops them in detail. The route has two broad halves. Chapters 0–10 construct the theory and its reasoning architecture; Chapters 11–20 test and specialize it; Chapters 21–22 extract the common design and identify its open frontier.

Design principle

Keep four questions separate while reading: What structure was declared? What obstruction was observed? What repair was authorized? What independent evidence admitted the change?

The view from above

LINCS begins with a model of the learning system itself. Objects and arrows describe its typed components and legal computations. Equations, cones, cocones, covers, and descent conditions declare the structural promises that those computations should satisfy. A failure of one of these promises produces a typed obstruction. Tangent structure asks how that obstruction changes under admitted local probes; quotienting removes variations that are irrelevant to the declared decision; localization assigns the surviving failure to a cover; repair proposes a typed change; and admission determines whether the maintained system should actually change.

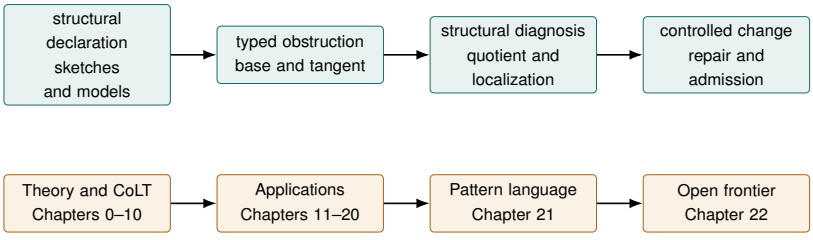


Figure 2: The upper sequence is the recurring LINCS computation. The lower route is the organization of the book.

The obstruction is the intellectual hinge. It is not defined by a scalar loss, although scalar observers may measure it. It does not identify

its own cause, although localization and domain assumptions may support a diagnosis. It does not authorize its own repair, although its type restricts the repair language. Finally, a successful local repair is not automatically admitted: finite behavior, uncertainty, invariants, and transfer must still be tested.

The foundations trail

The foundational chapters form a dependency graph rather than a list of independent topics.

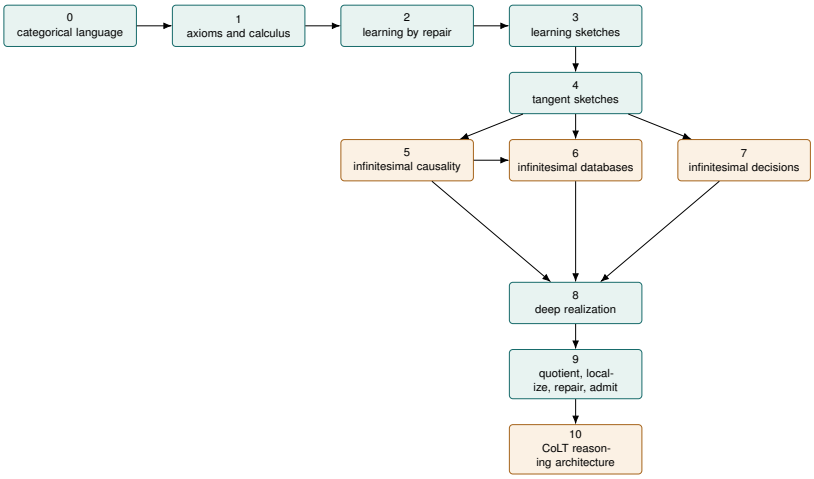
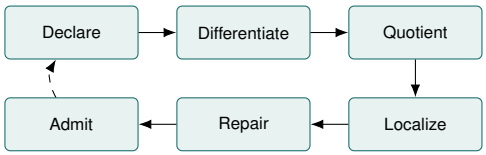


Figure 3: One recommended route through the foundations. Causality and decision making are distinct interpretations of tangent structure; their chapters communicate without being identified.

Chapters	Main object introduced	Question answered	Reader should retain
0–1	categorical language and universal obstruction	what is declared before a loss is chosen?	structure, observer, and scalarization have different types
2–3	six-stage workflow and learning sketch	how does a failed declaration become an operational signal?	factorization failure precedes diagnosis and repair
4	tangent learning sketch	how does a failure change under an admitted local probe?	infinitesimal non-compositionality is directional and typed
5–7	infinitesimal causality, databases, and decisions	when does tangent structure acquire causal, persistent, or decision semantics?	grounding, bracket enrichment, and decision relevance are additional declarations
8	differentiable realization	how can typed diagrams be compiled into layered learners?	architecture, differentiation, and optimization remain distinct
9	quotient, localization, repair, and admission	how is a structural change governed?	null variation, diagnosis, authority, and validation must be separated
10	CoLT reasoning state and transition record	how does governed repair become an auditable reasoning process?	only certified admission changes the maintained state

The recurring field procedure

Every application substitutes domain-specific objects into the same six-stage discipline.



Declare. State the objects, arrows, equations, universal properties, observers, and admissible probe language.

Differentiate. Ask how the obstruction moves under supported local variation; retain base and tangent information separately.

Quotient. Remove gauge, presentation, baseline, paraphrase, or other variation that the declared decision cannot observe.

Localize. Assign the surviving failure to modules, contexts, overlaps, skills, relations, agents, or argument fragments.

Repair. Search only within a registered language of typed changes and record what must remain invariant.

Admit. Use independent evidence to accept, reject, quarantine, or defer the proposal.

A chapter need not instantiate every stage with equal depth. The map is an audit device: a missing stage should be visible as an open obligation rather than silently replaced by a scalar objective.

The applications trail

The applications change the object of learning while preserving the structural grammar.

Chapter and domain	Structural focus	Control and admission lens
11 — causal discovery	visible intervention geometry and causal graphs; Lie-bracket nonclosure or unstable extraction	audit the model class and identification assumptions
12 — neural architecture	Kan-extended representation routes; disagreement between direct and extended base or tangent behavior	repair the route or architecture, then test finite behavior
13 — neural adaptation	ordered families of low-rank adapters; order-sensitive interaction	route, merge, separate, or retrain under an explicit admission test
14 — skills	sections over a Lie algebroid; anchor, bracket, or feasibility failure	repair the skill or workflow while preserving declared constraints
15 — reinforcement learning	Bellman and policy factorizations; base or tangent inconsistency	use decision quotients, effective sample size, quarantine, and recovery
16 — preference learning	relational preferences; cycles or a nonintegrable preference field	retain a relational fallback and guard any reward update
17 — relational learning	charts, joins, and shared manifold geometry; incidence, cocycle, or apex failure	make sparse, decision-preserving chart repairs
18 — distributed learning	local predictive sections; compatibility or effectivity failure	audit descent and apply transverse correction only when admitted
19 — argumentation	typed Toulmin roles and source-bound sections; role, qualifier, rebuttal, or gluing failure	localize edits while preserving source provenance
20 — foundation models	foundry artifacts and validation contracts; cross-agent or cross-stage failure	separate construction, scrutiny, and final admission

Chapters 11–15 emphasize causal, architectural, skill, and sequential decision systems. Chapters 16–20 emphasize relational evidence, distributed reasoning, argumentation, and foundation-model foundries. Chapter 21 then compresses their recurring choices into a pattern language. Chapter 22 asks how the framework might learn its own probes, covers, declarations, and theories, and how it might participate in scientific discovery rather than only repair a fixed model.

Choose a route

The book supports several routes besides reading every chapter in order.

Interest	Suggested route	Guiding question
Core theory	0–10, then 21	what is the minimal categorical contract for learning by repair?
Causality and discovery	0, 1, 3–6, 9–11, 22	when can a geometric obstruction support a causal or theoretical claim?
Decision making and RL	0, 1, 4, 7–10, 15–16	how does local variation alter universally extended decisions?
Deep architectures and skills	0, 3–4, 8–10, 12–14	how are compositional obligations compiled into trainable modules?
Relational and multi-agent reasoning	0, 3–4, 6, 9–10, 17–20	how do local structures glue without erasing disagreement or provenance?
Scientific creativity	1–7, 10–11, 21–22	when should repair change the model, and when should it change the theory?

Signposts for the journey

Three distinctions prevent most misreadings of the framework.

1. **Obstruction is not loss.** A scalar may observe or rank a typed failure, but it does not exhaust the failed structural promise.
2. **Diagnosis is not authority.** A localized obstruction may motivate a repair without identifying its cause or licensing a change.
3. **Infinitesimal coherence is not finite success.** Tangent agreement must be transported, tested, and admitted before it supports a deployment or scientific claim.

With those signposts in view, the long theoretical ascent has a practical destination. The applications do not merely illustrate category theory. They test whether one disciplined sequence—declare, differentiate, quotient, localize, repair, and admit—can make very different learning systems more inspectable and actionable.

Part I

Categorical Foundations

0

A Working Language of Compositionality

LINCS begins from a simple observation: a learning system is assembled from parts, and the ways those parts are composed carry meaning. An encoder feeds a predictor; an intervention changes a state; local models exchange information on overlaps; argument roles connect evidence to a claim. Category theory provides a language for saying which compositions are possible and which compositions are intended to agree.

This chapter develops the portion of that language used throughout the book. It is not a miniature survey of category theory. Its purpose is practical: after reading it, one should be able to identify the objects and arrows in a LINCS application, read its diagrams, understand what its sketch declares, and distinguish a structural obstruction from a numerical measurement of that obstruction.

The central reading habit is *type first*. Before asking for a formula or a loss, ask for the source and target of every map and for the two routes that are being compared.

0.1 Categories: typed composition

Definition 0.1 (Category)**.** A category $\mathcal{C}$ consists of:

1. a collection of objects $X, Y, Z, \ldots$;
2. for every pair X, Y, a collection of morphisms $f : X \to Y$;
3. an identity morphism $1_X : X \to X$ for every object; and
4. a composite $g \circ f : X \to Z$ whenever $f : X \to Y$ and $g : Y \to Z$.

Composition is associative, and identities are neutral:

$$h \circ (g \circ f) = (h \circ g) \circ f, \qquad f \circ 1_X = f = 1_Y \circ f.$$

The definition is deliberately spare. It says nothing about what an object "really is" or how a morphism is represented in software. That abstraction lets the same language describe several levels of a learning system.

Category	Objects	Morphisms	Composition means
Set	sets	functions	ordinary function composition
$\mathsf{Vect}_{\mathbb{R}}$	real vector spaces	linear maps	matrix or linear-map composition
Smooth	smooth manifolds	smooth maps	composition of differentiable transformations
A path category	nodes of a typed graph	directed paths	concatenating executable steps
A schema category	entity and relation types	declared relational paths	following foreign keys or joins
A category of models	model states or structured realizations	admissible transformations	transporting one realization to another

Table 1: The word *category* fixes a discipline of typed composition, not a single choice of mathematical object.

A MORPHISM IS NOT MERELY AN ARROW DRAWN ON PAPER. Its type $f : X \to Y$ is a contract. The composite $g \circ f$ exists only when the output type of f matches the input type of g. Many bugs in complex learning systems are already visible as type failures: an adapter expects a representation that an earlier adapter does not produce, a local database chart lacks the join witness needed by a downstream map, or a proposed argument repair changes a claim where only a warrant edit was authorized.

Example 0.2 (A skill pipeline)**.** In LASKO, a retrieval step $r : Q \to E$ maps a query to evidence, and a synthesis step $s : E \to A$ maps evidence to an answer. Their composite $s \circ r : Q \to A$ is a candidate skill. If a verifier $v : A \to V$ follows, then associativity guarantees that

$$v \circ (s \circ r) = (v \circ s) \circ r.$$

Associativity does not guarantee that the result is correct or executable; it guarantees that the two parenthesizations name the same typed composite. Additional declarations supply the substantive promises.

0.2 *Ordinary syntax and enriched semantics*

An ordinary category assigns a *set* of morphisms to each pair of objects. In many learning domains, however, the maps between two objects carry structure that matters to composition: they may form

a vector space, an ordered collection, a space with a distance, or a category of transformations. Enrichment makes that structure part of the categorical typing rather than attaching it afterward as an untyped score.

Definition 0.3 (Enriched category)**.** Fix a symmetric monoidal category $\mathcal{V}$, with tensor $\otimes$ and unit I. A $\mathcal{V}$-enriched category $\mathcal{C}$ has objects $X, Y, \ldots$, a hom-object

$$\mathcal{C}(X, Y) \in \mathcal{V}$$

for each pair, composition morphisms in $\mathcal{V}$

$$\mathcal{C}(Y, Z) \otimes \mathcal{C}(X, Y) \longrightarrow \mathcal{C}(X, Z),$$

and unit morphisms $I \to \mathcal{C}(X, X)$, satisfying enriched associativity and unit laws. A $\mathcal{V}$-functor maps objects and hom-objects while preserving this composition and these units.

Ordinary categories are precisely categories enriched over Set with Cartesian product. Other bases expose different information in the hom-objects:

Enriching category $\mathcal{V}$	What a hom-object records	Familiar case
Set	a set of maps	ordinary categories
the two-element ordered set **2**	whether a comparison holds	preorders
$([0,\infty], \geq, +, 0)$	a generalized directed distance	Lawvere metric spaces
$\mathsf{Vect}_{\mathbb{R}}$	a vector space of maps with bilinear composition	linear or differential operators
Cat	a category of maps and transformations between them	2-categories

Table 2: Enrichment changes the type of a hom from a bare collection into a structured object whose structure is respected by composition.

Boundary

Enriched is not a synonym for *equipped with additional structure*. A tangent category has a tangent endofunctor and coherence maps; a Markov category has symmetric monoidal, copying, and discarding structure; a site has a coverage; and a Lie algebroid has a bracket and anchor. Any of these may participate in an enriched construction, but none is automatically an enriched category merely by possessing that structure.

This distinction explains the convention adopted throughout the book. The ordinary category or sketch supplies a common combinatorial backbone of objects, paths, and factorization obligations. Each application then states the semantic structure actually used: enrichment of homs, tangent structure, monoidal structure, probability objects, sheaf semantics, Lie brackets, or some compatible combination. A norm, expectation, distance, order, or subtraction is available only after that extra structure has been declared.

Design principle

Read every LINCS model at two levels: first identify its ordinary typed compositional skeleton; then identify the enrichment or other semantic structure that makes its obstruction observable and its repairs admissible. Changing the second level can change the diagnostic without changing the underlying graph of compositions.

0.3 Graphs, paths, and diagrams

A directed graph lists objects and generating arrows but does not yet include all composites. Its *path category* $\mathrm{Path}(S)$ freely adds identities and finite paths. This distinction is useful in machine learning: the graph can describe primitive modules, while the path category describes every route that can be built from them.

Definition 0.4 (Diagram)**.** A diagram of shape J in a category $\mathcal{C}$ is a functor

$$D : J \longrightarrow \mathcal{C}.$$

Informally, J is the wiring pattern and D supplies its realization: each formal object receives an object of $\mathcal{C}$, and each formal arrow receives a morphism of $\mathcal{C}$.

Consider two adapters $A, B : H \to H$ acting on a hidden-state space. Their two orders form parallel paths

$$H \underset{A\circ B}{\overset{B\circ A}{\rightrightarrows}} H.$$

The paths are *parallel* because they share a source and target. This makes it meaningful to ask whether they agree. ALLORA does not assume that they do: the order discrepancy is part of the structure being diagnosed.

Definition 0.5 (Commutative diagram)**.** A diagram is commutative when every pair of parallel directed paths with the same endpoints has

the same composite. For the square

$$\begin{array}{ccc} X & \xrightarrow{f} & Y \\ {\scriptstyle u}\downarrow & & \downarrow{\scriptstyle v} \\ X' & \xrightarrow[f']{} & Y', \end{array}$$

commutativity means $v \circ f = f' \circ u$.

The square can express equivariance, simulation, consistency under a data transformation, or agreement between direct and transported prediction. Its meaning comes from the labels and chosen category; its categorical form says that two typed routes are intended to agree.

A diagram can be meaningful without commuting. In LINCS, a noncommutative diagram is often the observed candidate, while commutativity is the declared structural promise.

Example 0.6 (Bellman consistency in GIRL)**.** Let $\mathcal{V}$ be a space of value functions and let $P^\pi : \mathcal{V} \to \mathcal{V}$ be the Markov expectation operator for a policy π. With reward function R^π, define

$$B^\pi(V) = R^\pi + \gamma P^\pi V.$$

A parametrized value family $\widehat{V} : \Theta \to \mathcal{V}$ determines parallel maps

$$\widehat{V},\ B^\pi \circ \widehat{V} : \Theta \rightrightarrows \mathcal{V}.$$

Bellman consistency declares these maps equal. A sampled temporal-difference residual is a numerical observation of their failure to agree at the sampled state and transition.

0.4 *Functors: compositional translations*

Definition 0.7 (Functor)**.** A functor $F : \mathcal{C} \to \mathcal{D}$ assigns an object $F(X)$ of $\mathcal{D}$ to every object X of $\mathcal{C}$, and a morphism $F(f) : F(X) \to F(Y)$ to every morphism $f : X \to Y$, such that

$$F(1_X) = 1_{F(X)}, \qquad F(g \circ f) = F(g) \circ F(f).$$

The second equation is the important one: a functor preserves composition. It is therefore the natural mathematical form of a translation that respects how a system is assembled.

Three uses recur in this book.

Realization. A functor $D : J \to \mathcal{C}$ realizes a formal computational shape J as actual spaces and maps. In a neural realization, objects may be representation spaces and arrows may be differentiable modules. In RADAR, the shape can be a relational schema and the realization can assign local geometric charts.

Change of representation. A functor can translate a structured system into a different category without forgetting composition. A database instance is commonly treated as a functor from a schema category to Set [Spivak, 2012]. The schema says which relational paths exist; the functor supplies their tables, keys, and realized functions.

Differentiation. The tangent construction is a functor

$$T : \mathsf{Smooth} \longrightarrow \mathsf{Smooth}$$

that sends a manifold X to its tangent bundle TX and a smooth map f to its derivative-level map Tf. Functoriality is the chain rule:

$$T(g \circ f) = Tg \circ Tf.$$

This is why a declared compositional route can be lifted as a route rather than differentiated as an untyped scalar expression.

Design principle

Whenever the book writes $D : J \to \mathcal{C}$, read it in two passes. First read J as the declaration of available routes. Then read D as the candidate implementation of those routes. The distinction between shape and realization is essential to LINCS.

0.5 *Natural transformations: coherent change*

Functors describe structured systems. Natural transformations describe maps between such systems that are compatible with every arrow in the structure.

Definition 0.8 (Natural transformation)**.** Given functors $F, G : \mathcal{C} \to \mathcal{D}$, a natural transformation $\eta : F \Rightarrow G$ assigns to every object X a morphism $\eta_X : F(X) \to G(X)$ such that for every $f : X \to Y$,

$$\begin{array}{ccc} F(X) & \xrightarrow{F(f)} & F(Y) \\ {\scriptstyle \eta_X}\downarrow & & \downarrow{\scriptstyle \eta_Y} \\ G(X) & \xrightarrow[G(f)]{} & G(Y) \end{array}$$

commutes. Equivalently, $\eta_Y \circ F(f) = G(f) \circ \eta_X$.

The adjective *natural* means that the component maps do not act as unrelated patches. They transport coherently along every morphism of the source category.

Example 0.9 (A structured model update)**.** Suppose $D, D' : J \to \mathcal{C}$ are two realizations of the same learning sketch. A family of local updates

$\eta_j : D(j) \to D'(j)$ is a natural transformation only when it commutes with every realized operation $D(u)$. Thus an encoder update, predictor update, and decision update cannot be selected independently if the arrows connecting them are to retain their meaning.

This is stronger than the claim that every module improved on its local metric. It is one formal model for a *compositional repair*. LINCS does not require every repair to be a natural transformation, but when one is claimed, its naturality squares are explicit admission obligations.

0.6 *Reading coherent change in two dimensions*

Objects, functors, and natural transformations form successive levels: components, structured systems, and coherent changes of structured systems. Higher category theory continues this ladder, but the first two levels suffice for most constructions in this book.

There are two standard two-dimensional ways to depict the data of **CAT**. The first raises the directed diagrams of Section 0.3 by one categorical dimension. In the pasting scheme associated with Bénabou, categories are vertices, functors are directed edges, and a natural transformation is a 2-cell drawn between parallel functor edges [Bénabou, 1967]. This convention remains common in category theory texts; see, for example, Riehl [2017].

Street's string-diagram convention gives the planar dual picture [Street, 1996]. Natural transformations become vertices or boxes, functors become the wires incident on them, and categories become the regions separated by those wires. Vertical stacking expresses composition of natural transformations, while horizontal attachment expresses whiskering by a functor. Nothing mathematical has changed: the two drawings encode the same typed 2-cell, but make different dimensions visually prominent.

This second graphical language belongs to the calculational tradition developed by Marsden and subsequently presented systematically by Hinze and Marsden [Marsden, 2014, Hinze and Marsden, 2023]. It treats category theory itself diagrammatically, including natural transformations, adjunctions, monads, and Kan extensions. Cockett and Cruttwell specialize the calculus to tangent-category reasoning, using it to simplify calculations with tangent functors, lifts, flips, and Lie brackets [Cockett and Cruttwell, 2015]. The diagrams in this book specialize it once more: they form a LINCS dialect for displaying structural declarations, observers, obstructions, coherent repairs, and their transport.

Figure 0.6 compares the two conventions and then fixes the Street convention used in this book: functors point from the region on the left to the region on the right, and 2-cells are read from bottom to top. The first two panels depict the same 2-cell $\eta : F \Rightarrow G$. The third gives its first LINCS reading. A fixed-sketch repair $\rho : D \Rightarrow D'$ is not a collection of unrelated component edits; it is one coherent change between two

realization functors.

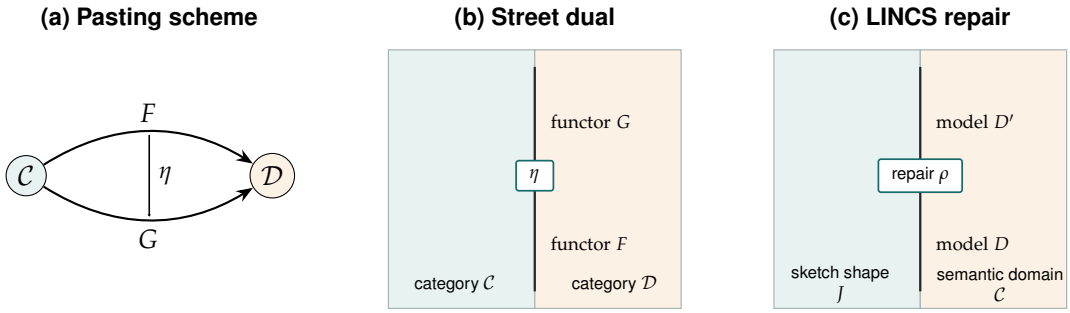


Figure 1: Two planar presentations of the same 2-cell, followed by its LINCS reading. The Street diagram is the planar dual of the pasting scheme: categories move from vertices to regions and the 2-cell moves from an arrow between parallel edges to a vertex on a wire. Region colors are pedagogical; the typing is carried by the category and functor labels.

This notation supplements rather than replaces the other diagram languages in the book. A commutative diagram displays obligations among objects and morphisms. A monoidal string diagram displays processes, tensor products, copying, and discarding. A **CAT**-diagram displays coherent changes of whole structured systems. The distinction matters: moving a box through a crossing is licensed only when the corresponding 2-categorical coherence law has been stated.

Design principle

Use graphical deformation only as a typed proof step. A visually smooth redrawing is not evidence of equality unless the relevant naturality, interchange, or tangent-coherence law licenses it.

0.7 Universal constructions

Some objects are characterized not by their internal representation but by how all other objects map to or from them. Such *universal properties* make constructions invariant under changes of implementation.

0.7.1 Products, pullbacks, and equalizers

A product $X \times Y$ comes with projections $\pi_X : X \times Y \to X$ and $\pi_Y : X \times Y \to Y$. For every pair $f : Z \to X$, $g : Z \to Y$, there is a unique map $\langle f, g \rangle : Z \to X \times Y$ whose projections recover f and g.

A pullback refines this idea. Given $f : X \to Z$ and $g : Y \to Z$, the pullback $X \times_Z Y$ represents pairs that agree after mapping to Z:

$$\begin{array}{ccc} X \times_Z Y & \xrightarrow{\pi_Y} & Y \\ {\scriptstyle \pi_X}\downarrow & & \downarrow{\scriptstyle g} \\ X & \xrightarrow[f]{} & Z. \end{array}$$

Every other commutative cone into this cospan factors uniquely through the pullback.

In RADAR, a pullback can express the relation obtained by joining two typed views over a shared key. A learned geometric apex is not merely an embedding with low pairwise error; it is asked to realize the declared incidence and join relations.

An equalizer of $f, g : X \rightrightarrows Y$ is an object $E \to X$ selecting exactly the part of X on which f and g agree. In SID, compatible local families form an equalizer of two restriction maps: one route restricts the first local section to an overlap, and the other restricts the second.

0.7.2 Coproducts, pushouts, and quotients

The dual constructions are defined by reversing arrows. A coproduct combines alternatives; a pushout glues objects along a shared part; a coequalizer identifies points or paths according to a declared relation. If $p, q : X \rightrightarrows Y$, a coequalizer is a map $c : Y \to Q$ satisfying $c \circ p = c \circ q$, universal among maps with that property.

This makes a quotient more than deletion. It records which distinctions are declared irrelevant. LINCS uses quotients to remove presentation changes, gauge directions, advantage baselines, paraphrases, or other variations that should not consume repair effort.

Construction	Structural question	LINCS reading	Example
Product	Can several observations be paired?	assemble jointly available information	a block of diagnostic probes
Pullback	Which pairs agree over a shared target?	enforce relational compatibility	database joins in RADAR
Equalizer	Where do parallel maps agree?	identify a consistency locus	compatible sections in SID
Coproduct	How are alternatives combined?	retain typed alternatives	families of repair proposals
Pushout	How are pieces glued along a common part?	construct a candidate global object	source-bound argument fragments
Coequalizer	Which distinctions are identified?	form a declared quotient	null or gauge variation

Table 3: Universal constructions used as structural declarations in LINCS.

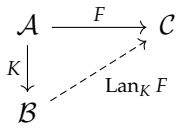
Boundary

A numerical optimizer may approximate a universal construction, but a low penalty is not itself the universal property. Existence, compatibility, and uniqueness obligations must be stated at the structural level and then connected to measurable tests.

0.8 Adjunctions and Kan extensions

An adjunction packages a best or universal translation between two categories. Functors $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{D} \to \mathcal{C}$ form an adjunction $F \dashv G$ when maps $F(X) \to Y$ correspond naturally to maps $X \to G(Y)$. The correspondence does not say that F and G are inverses. It says that one construction is universally adapted to mapping into or out of the other.

Kan extensions apply this idea to diagrams. Suppose $K : \mathcal{A} \to \mathcal{B}$ changes an indexing shape and $F : \mathcal{A} \to \mathcal{C}$ is known. A left Kan extension $\mathrm{Lan}_K F : \mathcal{B} \to \mathcal{C}$ is the universal way to extend F along K:



together with a natural comparison

$$\eta : F \Longrightarrow (\mathrm{Lan}_K F) \circ K.$$

The triangle therefore commutes up to the displayed natural transformation; it is not a strict equality unless additional hypotheses impose one. A right Kan extension is the dual universal construction, equipped with a comparison

$$(\mathrm{Ran}_K F) \circ K \Longrightarrow F.$$

In LINCS-KET, direct computation and Kan-extended computation provide two routes. Their discrepancy can be evaluated at the base level and after tangent lift. The Kan extension is not decorative terminology: it specifies which extension counts as canonical relative to the declared change of indexing shape.

Example 0.10 (Extending from observed contexts)**.** Let $\mathcal{A}$ index contexts in which a representation has been directly observed, and let $\mathcal{B}$ include additional composite contexts. The map $K : \mathcal{A} \to \mathcal{B}$ records the inclusion. A Kan extension constructs values on $\mathcal{B}$ from the observed diagram. A separately learned direct route on $\mathcal{B}$ can then be compared with this universal extension. LINCS asks whether the two routes agree and how their failure changes under admissible perturbations.

0.9 Sketches: declarations before objectives

A graph says what can be composed. A *sketch* additionally marks the diagrams that must commute and the cones or cocones that must satisfy universal properties [Ehresmann, 1968, Barr and Wells, 1999].

Definition 0.11 (Learning sketch)**.** In the notation used throughout this book, a learning sketch is

$$\mathbb{S} = (S, \mathcal{D}, \mathcal{L}, \mathcal{K}),$$

where S is a graph of formal operations, $\mathcal{D}$ is a family of equations between parallel paths, $\mathcal{L}$ is a family of designated limit cones, and $\mathcal{K}$ is a family of designated colimit cocones.

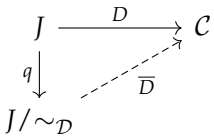
When the homs themselves carry essential structure, the same declaration can be made internally to an enriching category. A *$\mathcal{V}$-enriched learning sketch* consists of a small $\mathcal{V}$-category of generators and composites together with enriched equations and designated weighted limits and colimits. A model in a $\mathcal{V}$-category $\mathcal{C}$ is a $\mathcal{V}$-functor into $\mathcal{C}$ that preserves the designated weighted constructions. The ordinary definition above is the case $\mathcal{V} = \mathsf{Set}$.

The book retains the lighter notation $\mathbb{S}$ for both cases and names the enriching base whenever it affects a theorem, obstruction, observer, or repair. This prevents a metric, linear, probabilistic, or order structure used by an implementation from being mistaken for part of every learning sketch.

Let $J = \mathrm{Path}(S)$. The equations in $\mathcal{D}$ generate an equivalence relation on paths that is compatible with composition, and hence a quotient category

$$q : J \longrightarrow J/{\sim_{\mathcal{D}}}.$$

A candidate realization is a functor $D : J \to \mathcal{C}$. It satisfies the path declarations precisely when it factors through the quotient:



$$D = \overline{D} \circ q.$$

The dashed arrow is therefore a witness of compositionality. For a strict quotient q, such a factorization is unique when it exists.

This factorization is the formal hinge of LINCS. A candidate model need not satisfy it. Diagnosis may reveal a missing path factorization, a missing or nonunique universal filler for a declared cone or cocone, or a witness that is unstable under the declared probes. These

distinct failures become structural problems from which obstructions are derived. Only afterward are they observed through vector residuals, norms, likelihoods, hypothesis tests, or decision functionals.

Example 0.12 (One grammar, several applications)**.** The same four-part declaration takes different meanings later in the book:

- in ALLORA, paths are adapter orders and the declaration specifies which compositions should agree or remain safely compatible;
- in GIRL, a path equation expresses Bellman consistency;
- in RADAR, designated cones encode relational incidence and a shared geometric apex;
- in SID, an equalizer cone defines compatible local predictive states;
- in LINCS-Toulmin, arrows connect typed claim, ground, warrant, qualifier, rebuttal, and source roles.

The categorical grammar is shared; the semantics and admissible repairs are not.

0.10 Presheaves, sheaves, and local-to-global reasoning

Many LINCS applications distribute information across overlapping contexts. A presheaf records what is known locally and how it restricts to smaller contexts.

Definition 0.13 (Presheaf)**.** Let $\mathcal{U}$ be a category of contexts and inclusions. A presheaf on $\mathcal{U}$ with values in $\mathcal{C}$ is a contravariant functor

$$X : \mathcal{U}^{\mathrm{op}} \longrightarrow \mathcal{C}.$$

For an inclusion $V \subseteq U$, it supplies a restriction map $\rho_V^U : X(U) \to X(V)$. Functoriality says that restricting in stages agrees with restricting directly.

For the ordinary sheaf condition, take $\mathcal{C} = \mathbf{Set}$ and equip $\mathcal{U}$ with a specified coverage or Grothendieck topology. Suppose U is covered by contexts U_i. A family $x_i \in X(U_i)$ is *compatible* when its restrictions agree on every overlap:

$$\rho^{U_i}_{U_i \cap U_j}(x_i) = \rho^{U_j}_{U_i \cap U_j}(x_j).$$

A sheaf requires every compatible family to arise from a unique global section $x \in X(U)$. Compatibility is local agreement; *effectivity* is the existence of the global realization.

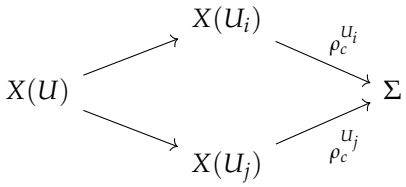
$$X(U) \rightrightarrows \prod_i X(U_i) \rightrightarrows \prod_{i,j} X(U_i \cap U_j)$$

The paired arrows compare restrictions to overlaps. In familiar set-valued cases, global sections form an equalizer of those arrows.

Example 0.14 (A crossword as a sketch and a sheaf)**.** A crossword makes the local-to-global structure visible without any categorical machinery. Let U_i be an across or down slot and let $X(U_i)$ be the set of candidate words licensed by its clue and length. If slots U_i and U_j meet at a square c, restriction extracts the letter occupying that square:

$$\rho_c^{U_i} : X(U_i) \longrightarrow \Sigma, \qquad \rho_c^{U_j} : X(U_j) \longrightarrow \Sigma,$$

where Σ is the alphabet. The local answers are compatible exactly when the two extracted letters agree.



For example, an across candidate CAT restricted at its second square and a down candidate ARE restricted at its first square both produce A. Replacing the down candidate by ORE produces O; the failure is localized to that crossing. A family of answers that agrees at every crossing glues to a unique completed grid relative to those local answers.

The same puzzle can be read as a sketch. Word slots, crossing cells, letters, and clue-licensed answer sets are typed objects; letter extraction supplies the arrows. The declaration says that the two routes into every crossing must agree, while a designated limit identifies a completed grid with the compatible family of its word-level restrictions. The sketch states the rules of the puzzle; the sheaf organizes its local solutions and their gluing. A wrong crossing is therefore a small, typed obstruction rather than a scalar score saying only that the grid is bad.

Example 0.15 (SID)**.** Each foundry maintains a local predictive state on its own context. Two foundries may agree on every audited pairwise overlap without their family belonging to the admitted class of global predictive models. SID therefore separates compatibility from effectivity and reports which part of the descent declaration fails.

Example 0.16 (LINCS-Toulmin)**.** A source passage may support a ground locally, a warrant may connect that ground to a claim, and

a qualifier may restrict the claim's scope. Agreement of isolated text spans is not enough. Their typed roles, source bindings, and restrictions must glue to a coherent argument. A failed warrant or incompatible qualifier is localized to a different part of the sheaf diagram and licenses a different repair.

Boundary

For an ordinary set-valued sheaf, compatibility on all pairwise overlaps is precisely the gluing hypothesis and already entails coherence on triple overlaps. Apparent higher-order obstructions arise when not all overlaps are observed, when the assignment is only a presheaf, when the admitted global model class is smaller than the sheaf semantics, or in enriched, derived, or higher-categorical versions where genuine cocycle data remain.

0.11 *Tangent structure and infinitesimal language*

LINCS also asks how a compositional failure changes under small, admissible perturbations. On smooth spaces, a tangent vector $v \in T_x X$ represents a first-order direction at x. A smooth map $f : X \to Y$ transports it by

$$Tf(x, v) = \big(f(x), Df(x)[v]\big).$$

The chain rule makes T functorial. A tangent category abstracts this behavior beyond ordinary smooth manifolds [Cockett and Cruttwell, 2014b].

Example 0.17 (An equivariant encoder and its tangent)**.** Let $a_g : X \to X$ transform an input, let $F_\theta : X \to Y$ be a learned encoder, and let $b_g : Y \to Y$ be the corresponding action on its representation. Equivariance is the declaration that the square

$$\begin{array}{ccc} X & \xrightarrow{F_\theta} & Y \\ {\scriptstyle a_g}\downarrow & & \downarrow{\scriptstyle b_g} \\ X & \xrightarrow[F_\theta]{} & Y \end{array} \qquad F_\theta \circ a_g = b_g \circ F_\theta$$

commutes. For an image encoder, a_g might rotate the image and b_g might rotate or reindex its feature representation. The sketch records this promise before any norm is chosen to measure its failure.

Applying T transports the entire declaration:

$$TF_\theta \circ Ta_g = Tb_g \circ TF_\theta.$$

In ordinary coordinates, its directional part is

$$DF_\theta(a_g x)\, Da_g(x)v = Db_g(F_\theta(x))\, DF_\theta(x)v.$$

The base square asks whether the two routes agree at x. The tangent square asks whether their first-order responses agree around x. A model can therefore fit sampled transformed inputs while remaining locally unstable to nearby perturbations.

An LLM encoder or hidden-state map can occupy the role of F_θ, but the transformation must be declared carefully. A self-attention block with no positional or causal structure is permutation equivariant; a standard causal LLM with positional information and a causal mask is not equivariant under arbitrary token permutations. Appropriate declarations may instead concern a controlled paraphrase, a formatting transformation, or a simultaneous reindexing of token representations and their structural data. The output action b_g may be the identity for a decision that should be invariant, or an explicit transport for token-level states. In either case, equivariance is a typed, testable promise, not an automatic property of the language model.

Design principle

A tangent category packages the chain rule so that LINCS can differentiate a compositional declaration, not merely the individual functions appearing in it.

Two constructions that look similar must be kept distinct. Write

$$T_n X = \underbrace{TX \times_X \cdots \times_X TX}_{n \text{ factors}}$$

for the fiber power of tangent vectors based at the same point, whereas $T^n X = T(T(\cdots T(X)))$ denotes iteration of the tangent functor. Thus $T_2 X = TX \times_X TX$ is the domain of fiberwise addition, while $T^2 X$ contains tangent vectors to TX.

Definition 0.18 (Tangent category)**.** A tangent structure on a category $\mathcal{C}$ is a tuple

$$\mathbb{T} = (T, p, 0, +, \ell, c)$$

with the following data and axioms.

1. $T : \mathcal{C} \to \mathcal{C}$ is an endofunctor and $p : T \Rightarrow 1_{\mathcal{C}}$ is a natural projection. Every finite fiber power $T_n X$ of $p_X : TX \to X$ exists, and every iterate T^m preserves these pullbacks.

2. Natural transformations

$$0 : 1_{\mathcal{C}} \Rightarrow T, \qquad + : T_2 \Rightarrow T$$

make $p_X : TX \to X$ a commutative monoid object in the slice $\mathcal{C}/X$. Hence only tangent vectors over the same base point may be added.

3. The vertical lift $\ell : T \Rightarrow T^2$, together with the zero section, is an additive-bundle morphism from $p_X : TX \to X$ to $T(p_X) : T^2X \to TX$.

4. The canonical flip $c : T^2 \Rightarrow T^2$ is an additive-bundle morphism from $T(p_X) : T^2X \to TX$ to $p_{TX} : T^2X \to TX$.

5. The lift and flip satisfy the tangent coherence axioms: in particular

$$c \circ c = 1_{T^2}, \qquad c \circ \ell = \ell.$$

   The standard iterated-lift and braid/interchange diagrams involving $\ell, c, T(\ell)$, and $T(c)$ also commute.

6. Let $\pi_0, \pi_1 : T_2X \rightrightarrows TX$ be the fiber-power projections and define the derived lift

$$\nu_X = T(+_X) \circ \langle \ell_X \circ \pi_0,\, 0_{TX} \circ \pi_1 \rangle : T_2X \longrightarrow T^2X.$$

   The vertical-lift universality axiom requires the square

$$\begin{array}{ccc} T_2X & \xrightarrow{\nu_X} & T^2X \\ {\scriptstyle p_X \circ \pi_0}\downarrow & & \downarrow{\scriptstyle T(p_X)} \\ X & \xrightarrow[0_X]{} & TX \end{array}$$

   to be a pullback. Equivalently, a second-order tangent that is vertical in the required sense factors uniquely through the derived lift.

A tangent category is a pair $(\mathcal{C}, \mathbb{T})$ of a category and a chosen tangent structure on it.

The last two clauses abbreviate the standard coherence diagrams and universality square of Cockett and Cruttwell [Cockett and Cruttwell, 2014b, Definition 2.3]; they are axioms, not consequences of the first four pieces of data. In local coordinates on a smooth manifold, the structure has the familiar form

$$\begin{aligned} p(x,v) &= x, & 0(x) &= (x,0), \\ +(x,v,w) &= (x, v+w), & \ell(x,v) &= (x,0;0,v), \\ c(x,v;\dot{x},\dot{v}) &= (x,\dot{x};v,\dot{v}). \end{aligned}$$

The flip exchanges the two first-order directions in a double tangent, while the vertical lift embeds a tangent vector as a vertical second-order direction.

Boundary

An endofunctor called T, or the availability of automatic differentiation, does not by itself make a category tangent. The fiber pullbacks, additive bundles, lift, flip, coherence diagrams, and vertical universality property are all part of the declaration. A LINCS application using only some of this structure should state that weaker assumption rather than silently claiming a tangent category.

There is a second boundary that is just as important for DLINCS. The category of finite-dimensional smooth manifolds and smooth maps is a tangent category, but it is not Cartesian closed. It has products, so a smooth family such as a motion of a body can be written

$$q : T \times B \longrightarrow E.$$

Cartesian closedness would additionally provide an exponential object E^T and the equivalent curried description

$$\widehat{q} : B \longrightarrow E^T.$$

The smooth path space E^T is generally infinite-dimensional and is not an object of the ordinary category of finite-dimensional manifolds. Thus tangent structure supplies differentiation, but it does not by itself supply internal function spaces [Lawvere, 1980].

The Kock–Lawvere solution is to embed ordinary smooth spaces into a suitable well-adapted *smooth topos* $\mathcal{E}$ [Kock, 2006, Moerdijk and Reyes, 1991]. Because a topos is Cartesian closed, it admits the natural correspondence

$$\mathcal{E}(P \times X, Y) \;\cong\; \mathcal{E}(P, Y^X).$$

A parametrized model $I : P \times X \to Y$ can therefore be treated internally as a map $\widehat{I} : P \to Y^X$ into a genuine model-space object. In the same setting an infinitesimal object D represents tangent structure on the appropriate infinitesimally linear or microlinear objects by $TX = X^D$. Cartesian closure and tangent structure remain logically distinct requirements; the smooth-topos setting is valuable because it supports them compatibly.

Design principle

Ordinary smooth manifolds motivate the tangent calculus of LINCS, but the internal higher-order semantics of DLINCS lives in a Cartesian-closed smooth topos. Products support parameterized evaluation; exponentials make spaces of models into objects of the theory.

This passage to a topos changes more than the available collection of smooth spaces. It also gives the theory an internal, generally intuitionistic logic. That logical consequence is the subject of Section 0.12.

0.12 *Internal logic: infinitesimals, subobjects, and interventions*

There is a shared logical feature behind two constructions used later in the book. Synthetic differential geometry represents first-order variation by the infinitesimal object

$$D = \{d \in R \mid d^2 = 0\},$$

while sheaf- and topos-based causal models represent predicates, admissible contexts, and domains of applicability by subobjects. In both settings the ambient category may have an *intuitionistic* internal logic: the law of excluded middle

$$\varphi \vee \neg\varphi$$

is not available as an unrestricted internal inference rule.

The reason is structural rather than philosophical. In Set, the subobject classifier is the two-valued set $\Omega = \{\bot, \top\}$, and every subset has a complement. In a general topos, a monomorphism $A \hookrightarrow X$ is still classified by a map $\chi_A : X \to \Omega$, but Ω is generally a Heyting algebra rather than a Boolean algebra [Mac Lane and Moerdijk, 1992]. Truth can depend on a stage or context, and a subobject need not have a complementary subobject. Thus failure to establish membership in A does not by itself establish membership in its complement.

For synthetic differential geometry this distinction is indispensable. Kock's axiom says that every map $g : D \to R$ has a unique form

$$g(d) = g(0) + d\,b.$$

If equality on D were decidable internally, excluded middle would permit the discontinuous case split "$d = 0$ or $d \neq 0$." Together with the Kock–Lawvere axiom, that split collapses the intended infinitesimal behavior and leads to contradiction [Kock, 2006, Section I.1]. The elements of D are therefore not ordinary hidden real numbers waiting to be classified as zero or nonzero; they are generalized elements whose first-order action is visible through maps out of D.

The causal parallel requires care. A causal intervention is not created merely by naming a subobject. In a categorical causal model it normally also requires a mechanism-replacement map, policy, or other intervention semantics. Subobjects instead describe such things as an observational event, an admissible context, the region on which a protocol is defined, or the stages at which an independence or identification

claim is validated. When these subobjects live in a non-Boolean sheaf topos, the model need not decide globally between "the causal claim holds" and "the causal claim fails." A claim may hold after restriction to a cover, remain unresolved at the present stage, or acquire a counterexample on a refinement. This is the logic used by intuitionistic j-do-calculus [Mahadevan, 2025c].

This common logic has four practical consequences for LINCS.

1. *Unknown is not false.* The absence of a factorization witness, causal certificate, or global section is not automatically a witness of nonexistence.

2. *Diagnosis is stage-sensitive.* An obstruction may be visible on one context, become locally trivial on a cover, or survive every admitted refinement. These are different judgments.

3. *Repair requires positive evidence.* A proposal is not admitted merely because its negation has not been proved. Admission must provide the declared witness, test, or certificate.

4. *Boolean reasoning is local and declared.* An application may use ordinary classical logic externally, and may reason classically about a decidable predicate or inside a Boolean model. What it may not do is assume without justification that every internal structural or causal predicate is decidable.

Design principle

LINCS distinguishes *not observed*, *observed to fail*, and *proved impossible*. Constructive internal logic preserves these three states until an observer or admission rule supplies enough evidence to identify them.

Example 0.19 (Holmes's rule of elimination)**.** In *The Sign of the Four*, Sherlock Holmes tells Watson, "when you have eliminated the impossible whatever remains, however improbable, must be the truth" [Doyle, 1890, Chapter 6]. The maxim cleanly separates possibility from probability, but it hides two logical certificates.

Suppose an external investigator has a finite, exhaustive family of hypotheses

$$H_1 \vee \cdots \vee H_n$$

and constructive refutations of every hypothesis except H_k. Then H_k follows even intuitionistically: eliminating cases from an explicitly given disjunction does not itself require excluded middle. The stronger Holmesian move is to assume that the hypothesis family is exhaustive, that each impossibility judgment is sound, and that no unrepresented explanation remains.

Internally, those assumptions may fail. If eliminating $\neg H$ yields only $\neg\neg H$, intuitionistic logic does not in general permit the final step $\neg\neg H \Rightarrow H$. Likewise, failure to construct H_i is not a construction of $\neg H_i$. For LINCS, Holmes's maxim is therefore a valid admission rule only when exhaustiveness and elimination are themselves certified; otherwise the remainder is a candidate for repair, not yet the truth.

This is not a demand that implementations abandon classical arithmetic or Boolean control flow. The distinction is between the external metatheory in which a model is built and the internal language used to reason about its context-dependent objects. One may study a smooth or causal topos using ordinary classical mathematics while correctly refraining from inserting excluded middle into that topos's internal logic.

0.13 Symbolic and neural reasoning

The relation between symbolic and neural reasoning has been debated since the emergence of connectionist AI. Symbolic accounts emphasize explicit representations, compositional rules, and search over expressions [Newell and Simon, 1976]; connectionist accounts emphasize distributed representations and learned transformations [Smolensky, 1988]. Neural–symbolic systems often respond by placing a rule layer beside, before, or after a neural model [d'Avila Garcez et al., 2002]. LINCS suggests a different division of labor. It does not treat symbolic and neural reasoning as rival substances, nor does it identify their reconciliation with a particular two-module architecture.

Three roles must be distinguished.

1. A *structural declaration* specifies typed objects, generating arrows, admissible diagrams, equations, covers, and repair operations. This is the role most closely associated with symbolic reasoning.

2. A *computational realization* assigns actual representations and maps to that declaration. Neural networks, embeddings, vector fields, and learned stochastic maps can all serve as realizations.

3. An *internal reasoning discipline* determines which claims, witnesses, restrictions, and repairs are warranted inside the chosen category. In a smooth or sheaf semantics this discipline may be intuitionistic and context-dependent, even when the external implementation uses ordinary classical arithmetic.

The interface between these roles is a typed obstruction. If $\mathbb{S}$ is the declared sketch and $D_\theta : J \to \mathcal{C}$ is a neural realization, then

$$(\mathbb{S}, D_\theta) \longmapsto \mathcal{O}_{\mathbb{S}}(D_\theta) \longmapsto \text{localized typed repair.}$$

The obstruction is neither an ordinary symbolic contradiction nor merely a scalar neural loss. It records how the learned realization fails relative to a particular structural promise. An observer may subsequently turn part of that object into a loss or test statistic, but the scalarization does not define the failure.

Category theory supplies the grammar of declaration and realization; it does not make a neural representation symbolic by fiat. The typing map, observers, and admission criteria must all be supplied and validated.

This account also makes the symbolic side revisable. A failure can sometimes be repaired by changing parameters inside a fixed sketch. At other times the available declaration is itself inadequate: a new object, arrow, axiom, cover, or observer must be proposed and old evidence transported into the enlarged theory. LINCS therefore distinguishes *repair within a symbolic language* from *repair of the language*. The latter is a controlled form of theory change, not the execution of a permanently fixed rule base.

There is consequently no single universal "LINCS logic." The internal logic depends on the semantic category selected by an application. Deep LINCS uses parametrized differentiable maps to realize declared diagrams; infinitesimal causality uses tangent directions and intervention protocols to reason about local causal variation; CoLT uses constructive witnesses, restrictions, and admission certificates to govern reasoning traces. Their commonality is the declaration–realization–obstruction–repair pattern, not one shared stock of symbols or one neural architecture.

Design principle

LINCS treats symbolic structure as a revisable categorical declaration, neural computation as one family of its realizations, and learning as the diagnosis and repair of failures between the two.

If $D : J \to \mathcal{C}$ is a realized diagram, its tangent lift is

$$TD : J \longrightarrow \mathcal{C}.$$

The base diagram asks whether declared routes agree. The tangent diagram asks whether their transported first-order behavior agrees along a specified class of probes.

Example 0.20 (Adapter order)**.** For two smooth adapter actions A and B, the base comparison is between $B \circ A$ and $A \circ B$. Their tangent lifts compare

$$T(B \circ A) = TB \circ TA, \qquad T(A \circ B) = TA \circ TB.$$

A base discrepancy measures finite order sensitivity. A tangent discrepancy asks how that sensitivity changes locally around the current representation and parameters.

Vector fields introduce another infinitesimal operation. The Lie bracket $[X, Y]$ is the infinitesimal commutator: it records the leading mixed-order discrepancy between the short flows generated in the two orders X then Y, and Y then X. It supports different semantics in different applications:

- BRIDGE/SKFM uses bracket closure as a falsifiable screen for a candidate intervention distribution;
- infinitesimal causality studies when intervention fields and their interactions can be given causal meaning;
- ALLORA uses order interactions among adapter-induced directions; and
- LASKO uses Lie-algebroid structure to test closure and executability of skill compositions.

Automatic differentiation supplies derivatives. It does not supply their semantics. A LINCS tangent site declares which points, directions, transports, covers, and null variations are admissible.

0.14 *From structural failure to learning signal*

We can now state the transition that motivates the book. Let a sketch $\mathbb{S}$ declare a structural promise and let D be a candidate model. Write

$$\mathrm{Fact}_{\mathbb{S}}(D)$$

for the space, category, or type of witnesses that D satisfies the declaration. Under the representability and triviality axiom introduced in Chapter 1, LINCS associates a base obstruction object

$$\mathcal{O}_0(D) = \mathcal{O}_{\mathbb{S}}(D),$$

which is trivial exactly when the declared factorization exists. After an admissible tangent lift, the corresponding infinitesimal obstruction is

$$\mathcal{O}_1(D) = \mathrm{INC}(D) = \mathcal{O}_{\mathbb{S}}(TD).$$

The obstruction object is intentionally abstract. Depending on the application, it may represent missing fillers, incompatible families, cohomology classes, route differences, failed closure conditions, or other typed diagnostics. To compute with it, one chooses an observer

$$\omega : \mathcal{O}_k(D) \longrightarrow Z$$

and perhaps a scalarization $\sigma : Z \to \mathbb{R}$. The composition $\sigma \circ \omega$ can become a loss or test statistic. It is evidence about the obstruction, not its definition.

Design principle

The order of construction is:

$$\text{typed declaration} \longrightarrow \text{factorization problem} \longrightarrow \text{obstruction} \longrightarrow \text{observation} \longrightarrow \text{scalar diagnostic}.$$

Reversing this order risks inventing a structural story for a convenient number after the fact.

A repair is similarly typed. It may change an adapter, a policy component, a relational chart, a local predictive section, or a Toulmin warrant. The application must say which changes are allowed, which variations are null, and what separate evidence admits the result.

0.15 A compact worked example

Consider a representation space H, two learned transformations $A_\theta, B_\phi : H \to H$, and a decision map $d : H \to Y$. Suppose the declaration says that the two transformations may be reordered without changing the decision:

$$d \circ B_\phi \circ A_\theta = d \circ A_\theta \circ B_\phi.$$

Category and diagram. The objects are H and Y; the arrows are the adapters, their composites, and d. The two sides are parallel paths from H to Y.

Sketch. The graph contains generators A, B, d, and $\mathcal{D}$ declares the two decision paths equal. The quotient of the path category identifies them.

Candidate model. The learned maps A_θ, B_ϕ, d define $D : \mathrm{Path}(S) \to \mathsf{Smooth}$. They may fail to factor through the quotient.

Observation. At a probe $h \in H$, one possible representative is

$$e(h) = d(B_\phi(A_\theta(h))) - d(A_\theta(B_\phi(h))).$$

This subtraction uses the additional assumption that Y has suitable linear or metric structure.

Tangent observation. For $v \in T_h H$, the derivative $De(h)[v]$ measures first-order change in the decision-level order discrepancy. Parameter probes $\dot\theta, \dot\phi$ answer a different question and must be labeled separately.

Quotient and localization. Directions that alter hidden coordinates but leave d invariant may be decision-null. The remaining discrepancy can be localized to layers, adapter blocks, data regimes, or token positions.

Repair and admission. A repair can adjust A_θ, B_ϕ, their routing, or the declared order relation. It is admitted only if the registered structural discrepancy improves on held-out probes while capability and safety constraints remain satisfied.

Admission contract

The equation above does not imply that all adapter orders ought to commute. It is a declared, decision-relative promise for a specified pair, regime, and support. If order carries intended semantics, forcing commutativity would be the wrong repair.

Application	Objects and arrows	Declaration	Obstruction	Typical repair
BRIDGE/SKFM	states and intervention fields	candidate distribution closes appropriately	bracket nonclosure or unstable causal extraction	revise fields, support, or downstream graph
LINCS-KET	indexed representations and attention routes	direct and extended routes agree	base or tangent route discrepancy	change architecture, route, or training rule
ALLORA	hidden states and adapters	declared orders preserve composition and safety	order-sensitive low-rank interaction	route, merge, separate, or retrain adapters
LASKO	task states and executable skills	sections compose and close	anchor, bracket, or feasibility failure	repair a skill or workflow edge
GIRL	states, values, and Bellman maps	value structure factors across computation	base or tangent Bellman failure	localized policy or representation update
LINCS-RLHF	alternatives and preference arrows	preference data support the chosen representation	cycle or non-integrable preference field	relational fallback or guarded reward update
RADAR	relational views, joins, and charts	local charts descend to a shared geometry	incidence, cocycle, or apex failure	sparse decision-preserving chart repair
SID	contexts, sections, and restrictions	local predictive states glue effectively	compatibility or global-realization failure	tangent preservation or transverse correction
LINCS-Toulmin	claims, grounds, warrants, and sources	typed argument roles restrict and glue	role, source, qualifier, or rebuttal failure	localized typed argument edit
Trustworthy foundation models	agent roles, evidence, claims, and obligations	typed foundry workflows pass local and global checks	unsupported synthesis or unsafe cross-role composition	repair evidence, role routing, or admission policy

Table 4: The formal language stays fixed while the application semantics change. Later chapters make each row precise.

0.16 How to read the rest of the book

Every chapter can be parsed with the same questions:

1. What category contains the realized system, and what enrichment or additional semantic structure is actually used?
2. What are the formal objects and generating arrows?
3. Which paths, cones, cocones, or descent data are declared?
4. What would witness the required factorization or universal property?
5. What kind of object records its failure?
6. Which base and tangent observations are actually measured?
7. Which directions are null, gauge, or decision-invariant?
8. Over what cover is the failure localized?
9. What is the typed repair language?
10. Which independent blocks admit, reject, or defer the repair?

Chapter 1 turns this grammar into an axiomatic contract and repair calculus; Chapter 2 makes it operational. Chapters 3 and 4 make the sketch and tangent constructions precise. The remaining foundations develop causal and decision interpretations, deep realization, localization, quotienting, and admission before the applications specialize the common language.

Further reading

For category theory, see Mac Lane [1998] and Riehl [2017]; for sketches and computational models, see Ehresmann [1968], Barr and Wells [1999], and Spivak [2012]. Enriched categories, enriched functors, and weighted universal constructions are developed systematically by Kelly [1982]. Local-to-global and tangent foundations are developed by Mac Lane and Moerdijk [1992], Cockett and Cruttwell [2014b], and Lee [2012]. For the two-dimensional graphical calculus used in this chapter, see Marsden [2014] and the systematic book-length account of Hinze and Marsden [2023]; its tangent-category specialization is exemplified by Cockett and Cruttwell [2015]. For a more extended discussion of categorical learning, see Fong et al. [2019], Cruttwell et al. [2022], and Gavranović et al. [2024].

1

Axioms of Structural Learning

LINCS AND A REPAIR CALCULUS

Chapter 0 supplied the working language of sketches, factorization, quotients, descent, tangent structure, repair, and admission. This chapter now states the structural contract of LINCS before the remaining foundations develop its ingredients one at a time. Its organization takes methodological inspiration from synthetic differential geometry. Kock begins that subject with an axiom saying that every map from the infinitesimal object D to the line R has a unique affine form [Kock, 2006]:

$$g(d) = g(0) + d\,b.$$

Equivalently, a canonical map $R \times R \to R^D$ is invertible. The axiom supplies a universal decomposition into value and infinitesimal slope; the subsequent theory explores what this single structural promise makes possible.

LINCS begins from an analogous organizing move, although not from the same algebraic axiom. Its primitive object is the *typed obstruction*: a universal carrier through which every admissible observation of a declared compositional failure must factor. This makes a failure prior to any chosen norm, loss, or coordinate system. Scalar losses become observers of the obstruction rather than definitions of it.

An axiomatization is not yet a calculus. Axioms give the semantic structures in which reasoning takes place. A calculus gives rules for transforming one licensed judgment into another. The distinction is familiar from causal inference. A structural causal model supplies semantics for interventions; do-calculus supplies rules for replacing expressions involving actions and observations when graphical separation conditions hold [Pearl, 2009b].

LINCS suggests a related, but more general, question:

> Under what declared conditions may one change the observer, quotient, localization, tangent level, repair target, or theory while preserving the meaning of a structural-learning judgment?

This chapter first states one universal obstruction axiom and then identifies the additional structural laws required to transport its meaning through tangent lift, presentation, localization, and repair. It then proposes a repair calculus as a disciplined interface for structural intervention. The rules are sound only relative to their stated sketch-theoretic hypotheses. They are not do-calculus rules for identifying causal effects, and no completeness theorem is claimed.

The word *calculus* is used in the proof-theoretic sense: a small collection of transformations on typed judgments. It is not another name for gradient calculus.

1.1 *What an axiomatization must distinguish*

Let

$$\mathbb{S} = (S, \mathcal{D}, \mathcal{L}, \mathcal{K})$$

be a learning sketch, let $J = \mathrm{Path}(S)$, and let

$$D : J \longrightarrow \mathcal{C}$$

be an admissible realization in a tangent category $(\mathcal{C}, T)$. The commutativity data induce

$$q_{\mathcal{D}} : J \longrightarrow J/{\sim_{\mathcal{D}}}.$$

The designated cones and cocones add limit and colimit obligations. We write $\mathrm{Fact}_{\mathbb{S}}(D)$ for the combined factorization problem.

Four levels must remain separate.

1. The *declaration* specifies what is required to compose.
2. The *realization* supplies a candidate model of that declaration.
3. The *observation* makes some aspect of failure detectable.
4. The *admission rule* decides whether evidence licenses a change.

Collapsing these levels makes an axiomatization circular. If a proposed repair may silently rewrite its observer, or if an observer may silently redefine the declaration, the system can remove evidence of failure without repairing the failure.

Boundary

The axioms define structural learning relative to a declaration. They do not declare the sketch correct, the cover complete, the observers sufficient, or the admitted policy normatively acceptable. Those are explicit modeling and validation commitments.

1.2 *The universal obstruction axiom*

For each pair $(\mathbb{S}, D)$, let $\mathcal{E}_{\mathbb{S},D}$ be a category of typed obstruction values with a distinguished trivial object $0_{\mathbb{S},D}$. No initial, terminal, or zero-object property is assumed unless a later law states it explicitly. For $Z \in \mathcal{E}_{\mathbb{S},D}$, write

$$\mathrm{Diag}_{\mathbb{S}}(D; Z)$$

for the set of admissible, structure-respecting diagnostics of the declared factorization problem with values in Z. Changing Z changes how the failure is observed; it must not silently change the declaration whose failure is being observed. For a morphism $u : Z \to Z'$, functoriality provides postcomposition

$$\mathrm{Diag}_{\mathbb{S}}(D; u) : \mathrm{Diag}_{\mathbb{S}}(D; Z) \longrightarrow \mathrm{Diag}_{\mathbb{S}}(D; Z').$$

Axiom 1.1 (Universal obstruction)**.** For every learning sketch $\mathbb{S}$ and admissible realization D, the diagnostic functor

$$\mathrm{Diag}_{\mathbb{S}}(D; -) : \mathcal{E}_{\mathbb{S},D} \longrightarrow \mathbf{Set}$$

is representable. Thus there is an obstruction object $\mathcal{O}_{\mathbb{S}}(D)$, unique up to unique isomorphism, and natural bijections

$$\mathrm{Diag}_{\mathbb{S}}(D; Z) \;\cong\; \mathrm{Hom}_{\mathcal{E}_{\mathbb{S},D}}\big(\mathcal{O}_{\mathbb{S}}(D), Z\big)$$

for every admissible observer Z. Moreover,

$$\mathcal{O}_{\mathbb{S}}(D) \cong 0_{\mathbb{S},D} \quad \Longleftrightarrow \quad \mathrm{Fact}_{\mathbb{S}}(D) \neq \varnothing.$$

Proposition 1.2 (Universal diagnostic factorization)**.** *Under the representing bijection, the identity of $\mathcal{O}_{\mathbb{S}}(D)$ determines a universal diagnostic*

$$\eta_{\mathbb{S},D} \in \mathrm{Diag}_{\mathbb{S}}\big(D; \mathcal{O}_{\mathbb{S}}(D)\big).$$

For every diagnostic $\delta \in \mathrm{Diag}_{\mathbb{S}}(D; Z)$, there is a unique $\widehat{\delta} : \mathcal{O}_{\mathbb{S}}(D) \to Z$ such that

$$\delta = \mathrm{Diag}_{\mathbb{S}}(D; \widehat{\delta})(\eta_{\mathbb{S},D}).$$

Proof. Define $\eta_{\mathbb{S},D}$ as the element corresponding to $1_{\mathcal{O}_{\mathbb{S}}(D)}$. Naturality of the representing bijection shows that postcomposing $\eta_{\mathbb{S},D}$ by $\widehat{\delta}$ corresponds to $\widehat{\delta} \circ 1_{\mathcal{O}_{\mathbb{S}}(D)} = \widehat{\delta}$. Taking $\widehat{\delta}$ to be the morphism corresponding to δ proves existence. Injectivity of the representing bijection proves uniqueness. □

The uniqueness belongs to this factorization of *diagnostics*. It does not imply that a repair exists, that an obstruction identifies its cause, or that two successful repairs must coincide.

Proposition 1.3 (Non-scalar primacy)**.** *Whenever $\mathbb{R}_{\geq 0}$ is an admissible observer object, every admissible structure-respecting scalar diagnostic for the declared failure is represented by a morphism*

$$\ell : \mathcal{O}_S(D) \longrightarrow \mathbb{R}_{\geq 0}.$$

Consequently, changing the scalarization changes the view of the obstruction, not the obstruction itself.

Proof. Apply Axiom 1.1 with $Z = \mathbb{R}_{\geq 0}$. Representability gives the unique morphism ℓ corresponding to the chosen scalar diagnostic. □

This is the book's central reversal of the ordinary optimization picture. The primitive failure is typed and potentially structured. A scalar may rank, threshold, or aggregate candidate repairs, but it can discard location, direction, interaction, provenance, and symmetry information carried by $\mathcal{O}_S(D)$.

Axiom 1.1 is a representability claim, not an observability theorem. The diagnostic functor may be poorly chosen or empirically incomplete. Observer adequacy remains a modeling commitment.

1.3 *Nonsmooth refinement: from a derivative to a response space*

The Kock–Lawvere axiom makes the first-order response unique. This is exactly the property that a nonsmooth component may lose. Let

$$\rho(x) = \max(0, x)$$

be the rectified linear unit. Away from the origin it has an ordinary derivative. At the origin its directional derivative is

$$\rho'(0; v) = \max(0, v),$$

which is positively homogeneous but not linear in v. There is therefore no single linear tangent map that describes its response in every direction. Relative to a sketch that declares unique first-order factorization, the kink is a compositional obstruction.

For a proper convex function $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$, convex analysis does not remove a kink by choosing an arbitrary derivative. It replaces the missing point-valued derivative at $x \in \operatorname{dom} f$ by the subdifferential

$$\partial f(x) = \{g \in \mathbb{R}^n : f(y) \geq f(x) + \langle g, y - x \rangle \text{ for every } y\}.$$

For ReLU,

$$\partial \rho(x) = \begin{cases} \{0\}, & x < 0, \\ [0, 1], & x = 0, \\ \{1\}, & x > 0. \end{cases}$$

The response is a singleton on the smooth strata and a convex family at the kink. Subdifferential calculus studies how such generalized responses transform under composition and other operations [Rockafellar, 1980].

Definition 1.4 (Registered generalized tangent response)**.** For an admissible nonsmooth component f at x, a generalized tangent response is registered data

$$\mathcal{R}_f(x)$$

whose elements are the first-order responses permitted by a named semantics, such as a convex or Clarke subdifferential or an automatic-differentiation conservative gradient. A computational selection is additional data

$$\sigma_f(x) \in \mathcal{R}_f(x).$$

The named semantics matters. A single convex ReLU has the convex subdifferential above, but the objective computed by an entire ReLU network is generally nonconvex. Clarke subdifferentials, limiting constructions, and conservative gradients need not agree, and their chain rules require different hypotheses. Modern nonsmooth automatic differentiation shows that backpropagation can efficiently transport conservative gradients through large classes of ReLU programs; it also shows that enumerating distinct Clarke subgradients can be computationally hard [Bolte et al., 2022]. Thus the value returned by a software derivative convention should not be silently identified with the whole generalized response object.

Design principle

At a nonsmooth point, distinguish three objects: the obstruction to a unique linear response, the registered space of admissible generalized responses, and the representative selected for computation. The second may serve as a repair language for the first; the third is an admission decision, not a mathematical inevitability.

This refinement provides the bridge to the next foundations. Tangent learning sketches in Chapter 4 need not assume that every component already has a unique classical derivative. They may instead declare which response object is admissible, test whether the selected responses compose, and retain the resulting failure as typed tangent information. DLINCS in Chapter 8 then becomes the computational layer that registers a selection policy, transports its choices through the model, and tests whether the resulting backward computation satisfies the declared tangent sketch.

A full categorical treatment would replace some point-valued tangent arrows by convex-valued correspondences, linear relations, or another suitable category of generalized responses. The present book does not assume that extension. It records the interface required of one: response semantics, selection, transport, and admission must remain explicit.

1.4 The semantic core

Definition 1.5 (INC category)**.** An INC category is a tangent category $(\mathcal{C}, T)$ equipped with:

1. a class of tangent learning sketches;
2. a class of admissible models $D : J \to \mathcal{C}$ for each sketch;
3. base factorization problems $\mathrm{Fact}_S(D)$;
4. transported tangent problems $\mathrm{Fact}_S(TD)$; and
5. when declared, an interaction signature Ω_D acting on an admissible tangent distribution $\mathcal{A}_D$.

The factorization definition of infinitesimal non-compositionality is

$$\mathrm{INC}(D) = \mathcal{O}_S(TD).$$

An interaction signature may additionally contain brackets, connection-based derivatives, curvature, canonical-flip comparisons, or higher jets. These refine tangent diagnosis; they do not replace the factorization problem.

Definition 1.6 (LINCS structure)**.** A LINCS structure on an INC category registers

$$\big(\mathrm{Fact}_S(D), \mathrm{Fact}_S(TD)\big)$$

is treated as the basic learning datum for every admissible model. When Ω_D is declared, the datum is refined by its associated closure or profile problems. Observation maps, scalarizations, repair languages, and admission contracts are additional registered structure.

This is deliberately a definition of *structure*, not yet of a new category. A category of LINCS systems additionally requires a specified class of objects—for example admissible pairs (S, D)—and morphisms that transport declarations, factorization problems, observers, and repair authority. Different applications choose different morphisms. The common structure therefore defines typed learning problems without pretending that one application-independent category of repairs has already been fixed.

1.5 Structural laws governing the obstruction

One axiom does not remove the need for hypotheses. Axiom 1.1 supplies the universal carrier of a declared failure and characterizes its vanishing. Tangent closure, presentation descent, interaction structure, and repair transport do not follow from representability alone. The following laws register that additional structure explicitly.

Corollary 1.7 (Base compositionality)**.** *An admissible model D satisfies its declared compositional structure exactly when its universal obstruction is trivial:*

$$D \models \mathbb{S} \quad \Longleftrightarrow \quad \mathrm{Fact}_{\mathbb{S}}(D) \neq \varnothing \quad \Longleftrightarrow \quad \mathcal{O}_{\mathbb{S}}(D) \cong 0_{\mathbb{S},D}.$$

Proof. The first equivalence is the definition of a model satisfying the learning sketch. The second is the triviality clause of Axiom 1.1. □

Structural law 1.8 (Tangent admissibility)**.** If D is an admissible model of a tangent learning sketch, then TD is again admissible, and its action agrees with the tangent structure declared by the sketch.

Structural law 1.9 (Presentation descent)**.** Suppose the implementation is expressed in a redundant presentation $\pi : P \to M$. Let $p_P : TP \to P$ and $p_M : TM \to M$ be the tangent projections. A proposed vector field $V_P : P \to TP$, satisfying $p_P \circ V_P = 1_P$, represents a vector field on the model object only if it descends to some $V_M : M \to TM$, with $p_M \circ V_M = 1_M$, such that

$$T\pi \circ V_P = V_M \circ \pi$$

or if a registered gauge fixing or equivariant horizontal lift supplies an equivalent descent certificate.

Corollary 1.10 (Tangent compositionality)**.** *Under tangent admissibility, Axiom 1.1 applies to TD. Define the infinitesimal obstruction by*

$$\mathrm{INC}(D) := \mathcal{O}_{\mathbb{S}}(TD).$$

Then infinitesimal compositionality is the same declared universal problem applied after tangent lift:

$$\mathrm{INC}(D) \cong 0_{\mathbb{S},TD} \quad \Longleftrightarrow \quad \mathrm{Fact}_{\mathbb{S}}(TD) \neq \varnothing.$$

Proof. Apply the triviality clause of Axiom 1.1 to the admissible realization TD. □

Structural law 1.11 (Sketch-specific interaction)**.** Every operation in a declared interaction signature Ω_D determines a typed closure, factorization, or profile problem on $\mathcal{A}_D$. Failure of that declared problem is admissible interaction data. No single operation, including the Lie bracket, is mandatory for every sketch.

Structural law 1.12 (Conditional obstruction transport)**.** A transport from base failure to an interaction-level obstruction,

$$\mathfrak{t}_{D,\mathbb{S}} : \mathcal{O}_{\mathbb{S}}(D) \rightsquigarrow \mathcal{O}_{\Omega,\mathbb{S}}(TD),$$

is additional sketch-relative data. It must be natural under admissible model transformations, descend through presentations, and preserve the semantics of the factorization problem. Its existence does not establish the empirical value of a scalarized tangent signal.

Structural law 1.13 (Functorial repair)**.** If $\alpha : D \to D'$ is an admissible repair, then $T\alpha : TD \to TD'$ is an admissible tangent repair. A repair advertised as compositionality-preserving must preserve the relevant factorization, and a repair advertised as local-to-global must commute with restriction up to the declared overlap coherence.

Proposition 1.14 (Trivial obstruction forces zero scalarization)**.** *Suppose $\mathcal{E}_{S,D}$ has zero morphisms and $0_{S,D}$ is a zero object. If $\mathcal{O}_S(D) \cong 0_{S,D}$, then every registered scalar observer*

$$\Phi_S : \mathcal{O}_S(D) \longrightarrow \mathbb{R}_{\geq 0}$$

is the zero morphism.

Proof. An object isomorphic to a zero object is initial. Hence there is exactly one morphism from $\mathcal{O}_S(D)$ to the scalar codomain. The zero morphism is one such morphism, so it is equal to Φ_S. □

The converse—that a zero scalar observer forces the obstruction to be trivial—requires the observer to reflect triviality and is therefore an additional identifiability condition.

Condition 1.15 (Observation specificity)**.** For an observation profile $\Psi_S(D, e)$, identification of a repair requires

$$\Psi_S(D, e) = \Psi_S(D, e') \quad \Longrightarrow \quad e \simeq e'.$$

Without this condition, the observations identify only an equivalence class or moduli space of repairs.

Admission contract

The registered repair language and structural laws make a repair judgment well typed. They do not make the repair beneficial. Admission requires evidence independent of the proposal signal and may assign zero weight to any base, tangent, or interaction observer that fails validation.

1.6 *From semantic axioms to a repair calculus*

Do-calculus transforms interventional distributions. Its rules are licensed by separation statements in a manipulated causal graph. A LINCS calculus has a different target. It transforms judgments about declared structures and their permitted repairs. A convenient judgment form is

$$\Gamma; \mathsf{S} \vdash e : D \Rightarrow D' \ [\mathcal{I} \mid \mathcal{E}],$$

where:

- Γ records semantic hypotheses, observers, and cover data;
- e is a registered edit from D to D';
- $\mathcal{I}$ lists the non-target obligations held invariant; and
- $\mathcal{E}$ records the evidence still required for admission.

The judgment says that e is a structurally licensed candidate. It does not yet say that e has been admitted or that it is a causal intervention on the represented domain.

When the registered edit is a natural transformation $e : D \Rightarrow D'$, the graphical language of Figure 0.6 makes an important distinction visible. The repair is a 2-cell between model functors. Applying an observer $\Omega : \mathcal{C} \to \mathcal{Z}$ does not create a new repair; it *whiskers* the existing one to obtain $\Omega e : \Omega D \Rightarrow \Omega D'$. The observed change is therefore downstream of the typed model change.

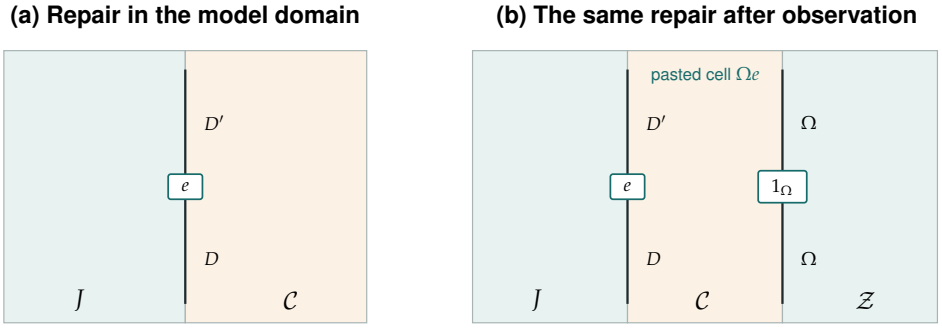


Figure 1.1: Horizontal pasting with the identity 2-cell on Ω transports a coherent repair to the observer domain. Observation can reveal or suppress features of a repair, but it does not change the repair's type.

If an edit changes the indexing sketch, its source and target models are not parallel functors and this picture no longer type-checks. Such theory-level repairs require a schema functor and a comparison cell, typically expressed through Kan extension; they are treated separately in the later chapters.

Comparison	Causal do-calculus	LINCS repair calculus
Semantic object	causal graph or structural causal model	learning sketch and its realization
Transformation target	observational and interventional distributions	structural judgments, observers, and repairs
Side condition	graphical separation after graph surgery	factorization, descent, naturality, invariance, or conservativity
Primary guarantee	equality of identified probabilistic expressions	preservation of a declared structural judgment
External requirement	causal assumptions and identification regime	observability, repair authority, and independent admission

Table 1.1: The analogy is methodological, not an identification of the two calculi.

1.7 *Six rule schemas*

The following rules form a candidate calculus. Each rule has a semantic side condition. Omitting the side condition turns a licensed transformation into an unsupported rewrite.

Rule R1: observer substitution. By Axiom 1.1, every observer is a map out of the same typed obstruction. Suppose $\omega : \mathcal{O}_S(D) \to Z$ and $\omega' : \mathcal{O}_S(D) \to Z'$, and suppose a registered isomorphism $i : Z \cong Z'$ satisfies

$$\omega' = i_{\mathrm{hom}} \circ \omega.$$

Then ω may be replaced by ω' in a diagnostic judgment, with its value transported along i. More general lossy observer changes require an explicit statement of which judgments they preserve. This rule changes how failure is seen, not what failure means.

Rule R2: presentation elimination. If the observer and proposed repair descend through $\pi : P \to M$, then a judgment on representatives may be replaced by a judgment on the quotient model:

$$\frac{T\pi \circ V_P = V_M \circ \pi \qquad \omega_P = \omega_M \circ \pi}{\Gamma; S \vdash e_P \implies \Gamma; S \vdash e_M}.$$

Gauge-dependent variation is thereby deleted from the learning judgment.

Rule R3: localization and gluing. Let $\{U_i \to U\}$ be a declared complete cover. A global diagnosis may be replaced by local diagnoses when the cover detects the relevant obstruction. Compatible local repairs may be replaced by a global repair only when descent is effective:

$$\left.\begin{array}{l} e_i|_{U_{ij}} \cong e_j|_{U_{ij}} \quad \text{with coherent descent data,} \\ \text{the registered descent problem is effective} \end{array}\right\} \implies \begin{array}{c} \exists\, e_U, \\ e_U|_{U_i} \cong e_i \end{array}.$$

Compatibility without effectivity licenses local agreement, not a global solution.

Rule R4: tangent transport. An admissible repair may be transported to the tangent level:

$$\frac{\Gamma; S \vdash \alpha : D \Rightarrow D'}{\Gamma_T; S \vdash T\alpha : TD \Rightarrow TD'}.$$

Here Γ_T is Γ transported through the registered tangent interface; its existence is part of the tangent-admissibility side condition. The reverse inference is not automatic. A tangent repair need not integrate to a finite base repair, and an interaction residual need not identify which base mechanism should change.

Rule R5: target surgery. Suppose e changes a registered target K and preserves every obligation in $\mathcal{I}$. Then one may reason in the edited realization $D[e/K]$ while retaining the non-target judgments:

$$D \models \mathcal{I} \text{ and } e \models \mathrm{Pres}(\mathcal{I}) \quad \Longrightarrow \quad D[e/K] \models \mathcal{I}.$$

This is the closest LINCS analogue of graph surgery. It is a structural intervention rule. It acquires Pearl-style causal semantics only when K denotes a grounded mechanism and the edit implements a realizable intervention protocol.

Rule R6: conservative declaration extension. Let $i : \mathbb{S} \hookrightarrow \mathbb{S}^+$ add objects, arrows, probes, or axioms. If i is conservative for the old language and the prior evidence transports along i, judgments expressed entirely in $\mathbb{S}$ remain licensed in $\mathbb{S}^+$:

$$\mathbb{S}^+ \models i(\varphi) \quad \Longleftrightarrow \quad \mathbb{S} \models \varphi \qquad (\varphi \text{ in the old language}).$$

This rule formalizes theory augmentation without silently rewriting earlier successes. It is central to abductive repair.

Appearance, reality, and declaration extension. McCarthy's distinction between appearance and reality gives R6 an epistemological interpretation [McCarthy, 2007]. Let

$$O : \mathcal{R} \longrightarrow \mathcal{A}$$

be an observation functor from a category of candidate realities to a category of appearances. Given appearance data $A : \mathcal{E} \to \mathcal{A}$, inference of a reality asks for a lift

$$\begin{array}{ccc} & & \mathcal{R} \\ & \overset{\widehat{A}}{\nearrow} & \downarrow O \\ \mathcal{E} & \xrightarrow[A]{} & \mathcal{A}. \end{array}$$

with $O \circ \widehat{A} = A$, or with a specified comparison when strict equality is not the intended semantics. The lift need not exist or be unique. If O is not faithful on the relevant structure, distinct realities may generate indistinguishable appearances. Additional views, actions, and experiments enlarge the probe family but do not guarantee identification.

When no object in the current declaration can support a satisfactory lift, R6 permits a new latent generator or law to be proposed in $\mathbb{S}^+$. Atomic theory is the canonical example: atoms were introduced as theoretical objects behind macroscopic appearances rather than read directly from them. Such an extension is admitted only if it transports established evidence, preserves successful old judgments, and supports independent consequences. The new vocabulary is a candidate explanation, not its own validation.

Design principle

A repair rule should state four things: what may change, what must remain invariant, how the change propagates, and which evidence is still required. The rule licenses a candidate transformation; admission remains a separate decision.

1.8 An admission rule is not an equational rule

The six schemas preserve structural judgments under explicit hypotheses. Admission has a different logical form. Let

$$\mathrm{Prop}(D, e)$$

denote the evidence used to propose e, and let $\mathrm{Val}(D, e; \mathcal{V})$ denote validation on evidence $\mathcal{V}$ excluded from proposal. Write $\mathcal{V} \perp \mathrm{Prop}(D, e)$ for the registered provenance condition expressing that exclusion. An admission rule may have the form

$$\frac{\Gamma; \mathsf{S} \vdash e : D \Rightarrow D'[\mathcal{I} \mid \mathcal{E}] \qquad \mathcal{V} \perp \mathrm{Prop}(D, e) \qquad \mathrm{Val}(D, e; \mathcal{V}) \models \mathcal{E}}{\Gamma, \mathcal{V}; \mathsf{S} \vdash \mathrm{Admit}(D')}.$$

This is not an equality transformation. It is a decision rule with provenance, independence, and threshold conditions. Keeping it outside the equational calculus prevents a system from confusing a well-formed edit with a successful one.

1.9 A minimal worked example

Suppose a sketch declares two routes

$$p, q : X \rightrightarrows Y, \qquad p \sim q,$$

and a realization D fails to factor through that equation. A parameter presentation $P \to M$ contains redundant coordinates, and a cover $\{U_i\}$ localizes the discrepancy.

The repair calculus organizes the reasoning:

1. apply R2 to remove parameter motion that does not descend to M;
2. apply R3 to diagnose the failure on the U_i;
3. propose a target edit of p and use R5 to preserve the declared behavior of every non-target path;
4. use R4 to test whether the repaired path remains coherent under admissible perturbations;

5. if the available repair language cannot resolve the obstruction, use R6 to propose a conservative extension $\mathbb{S}^+$; and

6. apply the admission rule on held-out observations before replacing D by D'.

No scalar loss is required for these transformations. A scalar observer may rank candidates, but it does not supply the descent, invariance, conservativity, or admission premises.

1.10 *Soundness, completeness, and the research frontier*

Proposition 1.16 (Conditional soundness of the rule schemas)**.** *Rules R1–R6 preserve their stated structural judgments in every LINCS realization satisfying their side conditions.*

Proof. R1 follows from Axiom 1.1 and transport along the registered isomorphism of observer codomains. R2 is presentation descent. R3 is the detection and effectivity property of the declared cover. R4 is tangent functoriality. R5 is precisely the invariant-preservation obligation attached to the target edit. R6 is conservativity of the sketch extension. Each conclusion therefore follows from a named semantic hypothesis rather than from the rule label alone. □

The proposition is intentionally relative. It does not show that the side conditions are empirically true, that the observer family is complete, or that every desirable repair can be derived. It is a schema-level soundness statement, not yet a metatheorem for one fixed formal syntax and semantics.

A genuine completeness theorem would require fixing:

1. a formal language of sketches, observers, and repairs;

2. a semantic equivalence relation on learning judgments;

3. an allowed class of quotients, covers, tangent transports, and theory extensions; and

4. a statement that every semantically valid transformation is derivable from a finite rule set.

That program is plausible for restricted fragments. For example, one could study finite sketches with effective finite covers, a fixed tangent signature, and a typed edit language. A universal calculus for arbitrary learning sketches is a much stronger claim and may require higher-categorical coherence rather than a small list of first-order rewrite rules.

Boundary

The proposed repair calculus does not identify causal effects from observational data. It axiomatizes controlled transformations of learning models. When a learning sketch contains a grounded causal model, its structural rules may be combined with do-calculus; they do not replace it.

1.11 *What this adds to the LINCS workflow*

The six-stage workflow introduced in the next chapter can be read at three levels:

1. Axiom 1.1 supplies the universal typed obstruction;
2. the *repair calculus* transforms licensed structural judgments; and
3. the *admission contract* decides whether a licensed candidate should alter maintained state.

This separation answers the question raised by the causal analogy in the Preface. LINCS inherits the lesson that action requires an explicit model and rule-governed surgery. Its distinctive contribution is to apply that lesson to the organization of learning itself: representations, computational routes, observers, covers, repair languages, architectures, and declarations become explicit objects of controlled intervention.

Chapter 2 now turns this contract into an operational workflow. The later foundations then justify its factorization, tangent, localization, and admission steps in detail.

Further reading

Kock exemplifies the strategy of beginning a differential theory from one generative axiom; LINCS borrows the organization, not the algebraic content [Kock, 2006]. A neighboring program explicitly called *Synthetic Machine Learning* develops a synthetic approach through categorical probability and statistics, Markov categories, information geometry, and noncommutative information geometry [Sepúlveda-Jiménez, 2025]. LINCS shares its synthetic ambition but takes declared compositional failure, typed obstruction, and controlled repair as its organizing primitives. Pearl develops do-calculus and the causal critique of association-only learning [Pearl, 2009b, 2018]. Sketches as categorical theories originate with Ehresmann [1968]; Barr and Wells [1999] connects them to categorical logic and model theory. Tangent categories supply the infinitesimal substrate [Cockett and Cruttwell, 2014b,

2017], while Riehl [2017] provides background on universal constructions, localization, and descent. Generalized gradients and nonsmooth backpropagation are treated by Rockafellar [1980], Bolte et al. [2022].

2
Learning by Repair

Machine learning usually begins with a scalar objective. A model makes a prediction, the objective reports an error, and optimization changes the parameters. This pattern is powerful, but it suppresses a prior question: *what structural promise did the model fail to keep?*

The LINCS viewpoint does not discard task loss. It asks which typed failure the loss is observing, and whether that failure licenses the proposed update.

In many systems the important promise is compositional. Two routes through a model should agree; local sections should glue; an update should preserve a decision; a preference field should integrate; a skill should remain executable after composition. A scalar loss may witness these failures, but it does not by itself preserve their type or location.

LINCS KEEPS THE FAILED DIAGRAM VISIBLE. A problem is declared as a learning sketch. Its candidate model is tested for a factorization, limit, colimit, descent, closure, or realization property. Failure produces an obstruction. The obstruction is then differentiated, quotiented by irrelevant variation, localized over a computational cover, and connected to a guarded repair.

2.1 *From loss minimization to structural diagnosis*

Let $J = \mathrm{Path}(S)$ be the path category generated by a sketch S, and let $D : J \to \mathcal{C}$ be a candidate diagram. Declared path equations induce a quotient $q : J \to J/\sim$. Compositionality is not defined by adding an arbitrary penalty; it is the existence of a factorization.

Definition 2.1 (Base obstruction)**.** The base obstruction $\mathcal{O}_0(D)$ records the failure of D to factor through the declared quotient, or to satisfy the specified universal property. Under Axiom 1.1, it is trivial precisely when the declared structural condition holds. A particular observer may fail to detect a nontrivial obstruction.

This formulation separates the categorical failure from its measurement. One may later observe $\mathcal{O}_0(D)$ through a norm, likelihood, test statistic, or decision functional, but those scalarizations are not the obstruction itself.

2.2 *An actionable model of the learning system*

Causal inference made a decisive methodological move: it replaced a purely associational description with a model whose mechanisms could be acted upon [Pearl, 2009b, 2018]. LINCS applies that lesson to the learner. The sketch does not merely describe the external domain; it can describe the architecture of learning, the transformations it performs, the obligations those transformations must satisfy, and the observations by which their failures become visible.

A minimal actionable learning model may be written

$$\mathfrak{L} = (\mathbb{S}, D, \Lambda, \mathcal{R}, \mathcal{A}),$$

where $\mathbb{S}$ is the structural declaration, D is its current realization, Λ is the observer family, $\mathcal{R}$ is the typed language of admissible interventions, and $\mathcal{A}$ is the independent decision rule. An edit $e \in \mathcal{R}(D)$ then induces

$$(\mathbb{S}, D) \xrightarrow{e} (\mathbb{S}_e, D_e) \xrightarrow{\mathrm{Obs}_\Lambda} \text{admission evidence.}$$

For a parameter update, $\mathbb{S}_e = \mathbb{S}$. For an architectural or theory-level repair, the declaration itself may change and the old model must transport conservatively into the new one.

This is analogous to causal surgery but intentionally more general. The edit must name its target and its invariants, yet it is causal only when the target is a domain mechanism and the edit has intervention semantics. Adapter replacement, reward-representation repair, cover refinement, and warrant revision are structural interventions on learning systems without thereby becoming causal effects.

Design principle

Optimization chooses among available edits. The actionable model determines what those edits mean, what they are allowed to change, and what must remain invariant. Declare the structural promise before selecting the scalar diagnostic. The same residual magnitude can represent different failures, and therefore authorize different repairs.

2.3 *The six-stage workflow*

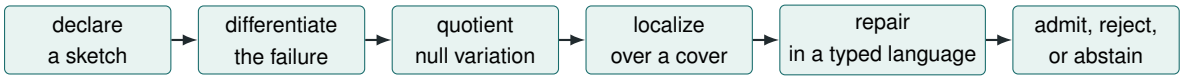


Figure 2.1: The recurring LINCS workflow. The arrows do not imply that every failure reaches admission: the process can stop at unidentifiability, incomplete localization, unsupported repair, or failed validation.

Declare. Specify the objects, arrows, path equations, universal properties, and observations that define success. This determines the type of failure the system is permitted to report.

Differentiate. Lift the sketch and its obstruction to a tangent setting. The infinitesimal obstruction

$$\mathcal{O}_1(D) = \mathrm{INC}(D)$$

describes how the structural failure moves under admissible perturbations.

Quotient. Remove variation that is presentationally redundant or null for the declared decision. A tangent direction that only changes a paraphrase, gauge, or advantage baseline should not masquerade as semantic repair.

Localize. Restrict the obstruction to a cover of layers, agents, argument roles, database faces, population strata, or computational blocks. Local agreement must still be distinguished from global realization.

Repair. Search only within a typed repair language: low-rank adapter changes, executable skill sections, policy updates, sparse relational corrections, or warrant-level argument edits.

Admit. Validate a candidate repair against separate structural, semantic, statistical, and operational blocks. Rejection and abstention are first-class outcomes.

2.4 *A small factorization example*

Suppose a system has an input object X, representation object H, decision object Y, and maps $h : X \to H$ and $d : H \to Y$. A declared input transformation $g : X \to X$ is represented by $\widetilde{g} : H \to H$ and $k : Y \to Y$. The intended square and decision-level equation are

$$\begin{array}{ccc} X & \xrightarrow{h} & H \\ {\scriptstyle g}\downarrow & & \downarrow{\scriptstyle \widetilde{g}} \\ X & \xrightarrow[h]{} & H \end{array} \qquad d \circ \widetilde{g} = k \circ d.$$

The two route differences can be observed numerically, but the declaration tells us whether the failure belongs to representation transport, decision equivariance, or both. Differentiating the square tells us which local directions change that failure to first order.

Proposition 2.2 (Infinitesimal candidate signal)**.** *If the declared square commutes at D, its tangent lift commutes along admissible tangent directions. Conversely, a nonzero tangent route difference identifies a first-order failure of the lifted declaration, not automatically a parameter update that will repair the base diagram.*

Proof. Functoriality of the tangent construction preserves the equality of the two composites. The converse claim is deliberately weaker: applying an observation map to the tangent route difference gives a diagnostic direction, while a compositional parameter repair additionally requires an identified map from parameter directions to diagram deformations. □

Boundary

A small infinitesimal obstruction does not establish semantic quality, and a large one does not prove that a supported repair exists. LINCS distinguishes diagnosis, localization, repair, and admission precisely to prevent those inferences.

2.5 *Repair is a gated transformation*

The following abstract procedure makes the separation operational.

LINCS diagnose-repair-admit cycle

Require: Learning sketch $\mathbb{S}$, candidate model D, cover $\mathcal{U}$, decision quotient χ, repair language $\mathcal{R}$
Ensure: Admitted model D', rejection, or abstention with certificate

Construct the typed base obstruction $\mathcal{O}_0(D) = \mathcal{O}_{\mathbb{S}}(D)$ through the registered diagnostic interface
Lift admissible probes and construct $\mathcal{O}_1(D) = \mathrm{INC}(D) = \mathcal{O}_{\mathbb{S}}(TD)$
Quotient presentation-null and decision-null variation using χ
Restrict the remaining obstruction over $\mathcal{U}$
if the cover is incomplete or the repair is unidentifiable **then**
    return abstention with missing-support certificate
end if
Generate candidate repairs $r \in \mathcal{R}$ from localized obligations
for all candidate repairs r **do**
    evaluate structural, semantic, statistical, and operational blocks
    if every required admission block passes **then**
        return $D' = r(D)$ with repair certificate
    end if
end for
return rejection with failed-block certificate

Admission contract

A repair may be admitted only if it improves the registered obstruction on held-out probes, preserves the declared decision semantics, stays within the allowed repair language, and passes application-specific safety or effectivity checks. Joint improvement in a single norm is not sufficient.

2.6 *One pattern, different domains*

The common architecture does not make the applications interchangeable. Each domain chooses a different sketch, obstruction, quotient, cover, repair language, and admission rule.

Application	Structural promise	Characteristic failure	Repair gate
BRIDGE/SKFM	Visible intervention fields support causal discovery	Lie-bracket nonclosure or unstable extraction	Falsified screen followed by an identified downstream learner
LINCS-KET	Direct and Kan routes agree	Base or tangent route incompatibility	Held-out, blockwise post-training admission
ALLORA	Adapter composition respects intended order	Noncommuting low-rank interventions	Capability and safety preservation
LASKO	Skill sections remain executable under composition	Anchor, bracket-closure, or feasibility failure	Executability and workflow validation
GIRL	Bellman structure factors across computation	Base or tangent Bellman obstruction	Layered policy admission and recovery
LINCS-RLHF	Pairwise preferences support the claimed representation	Cyclic or statistically unsupported scalar reward	Integrability test or relational fallback
RADAR	Private relational charts descend to a shared geometry	Incidence, cocycle, or apex incompatibility	Sparse, decision-preserving repair
SID	Compatible local predictive states realize globally	Compatibility or effectivity failure	Tangential preservation and transverse repair
LINCS-Toulmin	Typed argument roles glue across sources	Warrant, source, qualifier, or rebuttal failure	Typed structural repair certificate

Table 2.1: Nine application-specific realizations of the same LINCS pattern.

Design principle: What an empirical chapter must retain

An application chapter reports not only the best admitted result, but also the registered probes, failed repairs, incomplete covers, and boundary between the measured object and the broader claim. Negative results are part of the design evidence.

2.7 *What changes in the chapters ahead*

Part I develops learning sketches, tangent failures, infinitesimal causality, deep realization, quotienting, localization, repair, and admission. Application chapters vary the domain while retaining this spine; the final synthesis separates the reusable LINCS pattern language from domain-specific elements. In each case the sketch also functions as an actionable model of the learner: it determines where structural surgery can occur and which non-target obligations the repair must preserve.

Further reading

The categorical background begins with standard introductions such as Mac Lane [1998] and Riehl [2017]. For the presentation of mathematical theories by sketches, the classical line runs from Ehresmann's original program through categorical model theory and accessible categories [Ehresmann, 1968, Barr and Wells, 1999, Makkai and Paré, 1989, Adámek and Rosický, 1994]. These sources explain why a diagram can specify a theory before any numerical objective is chosen.

For readers interested in compositional learning itself, Fong et al. [2019] treats backpropagation functorially, while Cruttwell et al. [2022] develops categorical foundations for gradient-based learning. The more recent categorical-deep-learning program argues for an algebraic bridge between architectural constraints and their implementations [Gavranović et al., 2024]. LINCS is closest to this literature in its insistence on typed composition, but differs by making failed factorization and guarded repair the central learning event.

The interventionist contrast is developed by Pearl [2009b, 2018]. Those works motivate the need for models that support action rather than association alone. LINCS adopts that discipline at the level of the learning system while retaining a strict boundary between a general structural edit and a causally grounded intervention.

3
Learning Sketches and Factorization Obstructions

A loss tells us that a model is wrong by some amount. A learning sketch first tells us *what kind of agreement the model promised*. It records the objects and operations of the problem, the paths intended to agree, and any limit or colimit properties the candidate model must realize. This declaration turns non-compositionality into a typed factorization failure before a norm, likelihood, or empirical statistic is chosen.

Sketches originate in categorical logic and model theory. Here they serve as compact, machine-readable declarations of learning structure [Ehresmann, 1968, Barr and Wells, 1999].

3.1 From computational graphs to learning sketches

Let S be a directed graph of formal learning operations and let

$$J = \mathrm{Path}(S)$$

be its free category. Objects of J can denote data, representations, predictions, policies, local sections, or world-model fragments. Its arrows are formal composites of encoders, transitions, interventions, restrictions, updates, and decision maps.

A bare graph says which composites can be formed. It does not say which composites ought to agree. A learning sketch adds that declaration.

Definition 3.1 (Learning sketch)**.** A learning sketch is a quadruple

$$\mathbb{S} = (S, \mathcal{D}, \mathcal{L}, \mathcal{K}),$$

where:

- S is a graph of formal learning operations;
- $\mathcal{D}$ is a family of equations between parallel paths;
- $\mathcal{L}$ is a family of designated cones; and
- $\mathcal{K}$ is a family of designated cocones.

The equations in $\mathcal{D}$ generate a congruence $\sim_{\mathcal{D}}$ on $J = \mathrm{Path}(S)$ and hence a quotient

$$q_{\mathcal{D}} : J \longrightarrow J/{\sim_{\mathcal{D}}}.$$

The three kinds of declaration play different roles. Path equations express route agreement. Designated cones express reconstruction, synchronization, or shared-state conditions. Designated cocones express aggregation, identification, or gluing. Treating all three as a single scalar penalty would erase this distinction.

Design principle

Declare route equalities and universal properties separately. They can fail for different reasons, localize to different components, and require different repair languages.

3.2 *Sketches as categorical theories*

The sketch is more than a dataflow schema. It is a presentation of a categorical theory: generators provide the sorts and primitive operations, while path equations and designated universal constructions provide its axioms. This viewpoint continues the separation between syntax and semantics made explicit by Lawvere's functorial semantics of algebraic theories [Lawvere, 1963].

A single-sorted finitary Lawvere theory is a small category whose objects are finite powers of one distinguished object. Its arrows represent operations, composition represents substitution, and equality of arrows records equational axioms. A model in **Set** is a product-preserving functor. For example, a theory of monoids contains a multiplication $m : X^2 \to X$ and a unit $e : 1 \to X$; associativity and the unit laws are equalities between the corresponding composite arrows. A product-preserving functor interprets these generators as an actual monoid.

Sketches enlarge this language. They can use many sorts, select particular cones and cocones, and present structures whose axioms involve specified limits or colimits. A learning sketch uses the same division of labor: $\mathbb{S}$ is the theory, while a functor D is one attempted interpretation of it. Learning inside a fixed sketch changes the interpretation without silently changing the theory that the interpretation is supposed to satisfy.

Under standard size and arity hypotheses, categories of models of suitable limit theories and sketches are accessible or locally presentable. This makes a theory-generated model space amenable to construction from small presentable pieces [Makkai and Paré, 1989, Adámek and Rosický, 1994].

Definition 3.2 (Sketch augmentation)**.** A sketch augmentation is a morphism

$$\iota : \mathbb{S} \longrightarrow \mathbb{S}^{+}$$

that transports the existing generators and declarations while possibly adding new objects, arrows, path equations, designated cones, or designated cocones. It induces a restriction functor

$$\iota^{*} : \mathrm{Mod}(\mathbb{S}^{+}, \mathcal{C}) \longrightarrow \mathrm{Mod}(\mathbb{S}, \mathcal{C})$$

that forgets the added structure.

For an old model D, the fiber of ι^* over D is the space of its expansions to the augmented theory. An empty fiber says that the proposed theory is incompatible with D. An essentially unique expansion is characteristic of a definitional extension. Multiple inequivalent expansions expose genuine underdetermination: the new generators or axioms require evidence not contained in the old model.

Boundary

Repairing a model of $\mathbb{S}$ and augmenting $\mathbb{S}$ are different learning problems. The first searches for a better interpretation of a fixed theory. The second changes the space of admissible interpretations and therefore requires stronger admission tests.

3.3 *Models and the factorization test*

Let $\mathcal{C}$ be the category in which the learning system is realized. A candidate system assigns a realized object to every formal object and a realized computation to every formal path.

Definition 3.3 (Model of a learning sketch)**.** A candidate model of $\mathbb{S}$ in $\mathcal{C}$ is a functor

$$D : J \longrightarrow \mathcal{C}.$$

It is a strict model when there is a functor

$$\overline{D} : J/{\sim_{\mathcal{D}}} \longrightarrow \mathcal{C}$$

such that $D = \overline{D} q_{\mathcal{D}}$, and the images of the designated cones and cocones realize the universal properties specified by $\mathcal{L}$ and $\mathcal{K}$.

The commutativity component is therefore the lifting problem

$$\begin{array}{ccc} J & \xrightarrow{\ D\ } & \mathcal{C} \\ {\scriptstyle q_{\mathcal{D}}}\downarrow & \nearrow_{\overline{D}} & \\ J/{\sim_{\mathcal{D}}} & & \end{array}$$

rather than an arithmetic expression such as $g \circ f - f \circ g$. The latter expression becomes available only after choosing additive enrichment or an observation into a numerical space.

Definition 3.4 (Factorization problem)**.** For a candidate model D, define

$$\mathrm{Fact}_{\mathbb{S}}(D) = \left\{ \overline{D} \;\middle|\; D = \overline{D} q_{\mathcal{D}} \text{ and } \overline{D} \text{ realizes } \mathcal{L}, \mathcal{K} \right\}.$$

The model is compositional exactly when $\mathrm{Fact}_{\mathbb{S}}(D)$ is inhabited.

This set notation is economical, but the factorization problem can carry more structure: a category or space of fillers, a proof object, a sheaf of local fillers, or a derived obstruction object. LINCS leaves that choice to the realization.

Definition 3.5 (Base obstruction)**.** Subject to Axiom 1.1, the base obstruction is the representing object

$$\mathcal{O}_0(D) = \mathcal{O}_S(D)$$

for admissible diagnostics of the factorization problem. Its distinguished trivial state corresponds exactly to an inhabited factorization problem.

Boundary

The obstruction object is not assumed to be a vector. Depending on the application it may be a family of parallel-path witnesses, a cohomology class, a failure of effectivity, or a missing filler. A statistical hypothesis or test result is an observation of such an obstruction, not the obstruction itself.

3.4 *Three declarations, three characteristic failures*

Path agreement. For parallel paths $p, q : x \to y$, the declaration $p \sim q$ asks whether $D(p) = D(q)$. Equivariance, Bellman consistency, direct-versus-extended prediction, and argument-role transport all have this form.

Limit realization. A designated cone can say that a representation is reconstructible from compatible views, that a shared state is an equalizer, or that a database apex is a pullback. The characteristic failure is not merely disagreement of two arrows; it can be failure of existence or uniqueness of the mediating map.

Colimit realization. A designated cocone can describe empirical aggregation, quotient identification, or gluing of local pieces. Here the failure may be that local objects cannot be assembled into a global realization with the declared universal property.

3.5 *A running equivariance sketch*

Let a group element g act on inputs by $a_g : X \to X$ and on outputs by $b_g : Y \to Y$. A learned map $F_\theta : X \to Y$ is declared equivariant by the

Declaration	Question	Typical witness	Possible repair
Path equation	Do two routes agree?	Parallel-path incompatibility	Change an arrow, representation, or route
Limit cone	Is the global object determined by its views?	Missing or nonunique mediator	Add a view, constraint, or shared object
Colimit cocone	Do local pieces glue or aggregate?	Incompatible identifications or failed effectivity	Repair overlaps, quotient, or gluing data

Table 3.1: The sketch remembers which universal promise has failed.

square

$$\begin{array}{ccc} X & \xrightarrow{F_\theta} & Y \\ {\scriptstyle a_g}\downarrow & & \downarrow{\scriptstyle b_g} \\ X & \xrightarrow[F_\theta]{} & Y. \end{array}$$

The indexing graph contains the two parallel paths

$$F_\theta a_g, \qquad b_g F_\theta : X \to Y,$$

and $\mathcal{D}$ identifies them. The base obstruction is the failure of the realized diagram to factor through this quotient.

If X and Y are normed spaces, one possible observation is

$$E_{g,\theta}(x) = F_\theta(a_g x) - b_g F_\theta(x), \qquad \ell_{g,\theta}(x) = \|E_{g,\theta}(x)\|^2.$$

The vector $E_{g,\theta}(x)$ and scalar $\ell_{g,\theta}(x)$ are useful measurements, but neither defines equivariance. The declared square does.

Example 0.17 introduced this square using an image or language encoder. In the language setting, a_g must preserve or transform explicitly declared structure: it cannot be an arbitrary token permutation of a causal LLM. Depending on the task, b_g may reindex token-level states or act as the identity on an invariant decision. This choice belongs to the sketch and determines what an observed discrepancy means.

Example 3.6 (Bellman consistency)**.** Let $\mathcal{V}$ be a value-function space, $P^\pi : \mathcal{V} \to \mathcal{V}$ the Markov expectation operator, and

$$B^\pi(V) = R^\pi + \gamma P^\pi V$$

the Bellman operator. A parameterized value family $\widehat{V} : \Theta \to \mathcal{V}$ determines the parallel pair

$$\widehat{V},\ B^\pi \widehat{V} : \Theta \rightrightarrows \mathcal{V}.$$

Declaring this pair equal expresses Bellman consistency. Sampled TD and GTD objectives are scalar observations of its factorization obstruction; they do not replace the declaration [Sutton, 1988, Sutton et al., 2008, Sutton and Barto, 2018].

3.6 *Observation and scalarization*

A computation eventually needs measurements. LINCS separates their selection from the structural declaration.

Definition 3.7 (Observation of an obstruction)**.** An observation is a map

$$\omega : \mathcal{O}_0(D) \longrightarrow Z$$

into a space Z of inspectable witnesses. A scalarization is a further map $\sigma : Z \to \mathbb{R}$, possibly depending on data, probes, uncertainty, or an experimental design.

Different observations of the same obstruction can answer different questions. A supremum norm emphasizes worst-case disagreement; an empirical mean estimates average disagreement under a sampling law; a hypothesis test asks whether the available evidence distinguishes the obstruction from a null; a decision functional asks whether the failure changes an action.

Scalarization is not forbidden. It is postponed until the type, location, and evidential status of the failure are explicit.

Admission contract

A scalar diagnostic may guide repair only relative to its declared observation map, sampling regime, and support. A low value outside that contract does not establish that the categorical obstruction has vanished.

3.7 *Presentation, transport, and sketchability*

A sketch is itself a presentation. Two presentations can describe equivalent learning behavior while exposing different paths, auxiliaries, or parameter symmetries. Later chapters therefore quotient presentation-null variation before assigning repair effort.

Maps between learning graphs also require care. For unconstrained categorical schemas, functorial data migration is automatic. Once designated cones and cocones are added, a graph map need not preserve them [Spivak, 2014]. A change of learning sketch must state which equations, universal properties, admissible models, and repair spaces it transports.

Ordinary sketches have a further expressivity boundary: not every natural category of structured objects and structure-preserving morphisms is the category of set-valued models of a sketch [Barr and Wells, 1992]. Higher, enriched, fibrational, or homotopical presentations may be needed when admissibility itself quantifies over structure outside ordinary sketch logic. Section 0.2 fixes the book's convention: the ordinary sketch records the shared compositional skeleton, while

any hom-object enrichment or other semantic structure required by an application must be declared separately.

Boundary

LINCS is relative to a declared class of models and morphisms. It does not claim that every learning system has one canonical ordinary-sketch presentation.

3.8 *A sketch as a model of learning*

The declaration is not merely a convenient notation for a loss. It is the model that makes the learning process actionable. Its objects can represent data, representations, policies, modules, agents, arguments, or evidence states; its arrows represent the transformations through which learning is organized; and its equations and universal properties state which parts of that organization must remain coherent.

This shifts the role of modeling. In conventional causal inference, an SCM or SEM models mechanisms in an external domain and supports surgery on a designated mechanism. A learning sketch can include such a domain model, but it can also model the learner that constructs, compares, and repairs domain models. The resulting intervention target may be a parameter, a component morphism, an architectural route, an observation map, or the declaration itself.

The distinction matters because a repair cannot be licensed by location alone. A failed path identifies where the current declaration is not realized; it does not determine whether one should change a parameter, replace a mechanism, redraw the architecture, or revise the theory. Those choices belong to the repair language and admission contract attached to the sketch.

3.9 *The declaration card*

Before differentiation or optimization, a learning-sketch declaration should answer:

1. What are the formal objects and generating arrows?
2. Which parallel paths are required to agree?
3. Which cones and cocones are designated?
4. What counts as an admissible realized model?
5. What object records failure of the factorization problem?
6. Which observation maps make that failure measurable?

7. Which variations are merely presentational?
8. What evidence would justify a repair rather than abstention?
9. Which parts of the sketch may be augmented, and what would make such an extension conservative?
10. Which edits count as structural interventions, what do they target, and which non-target obligations must they preserve?

A failed factorization can reveal either of two gaps. The declared theory may be adequate while the candidate model realizes it poorly, or the persistent failure may indicate that the sketch itself lacks a generator, relation, or universal construction needed to express the phenomenon. The obstruction alone does not decide between these explanations. We must ask how it changes under admissible variations and where those changes are supported.

The next chapter supplies this sensitivity analysis by applying tangent structure to the declaration. The result is not simply a derivative of the final scalar loss: it is a lifted factorization problem whose obstruction records how compositional failure moves. At first this diagnoses gaps in a model of a fixed sketch. Later, the same localization logic will help identify explanatory pressure on the sketch and motivate a carefully admitted theory augmentation.

Further reading

Lawvere's functorial semantics makes models into structure-preserving functors from an algebraic theory [Lawvere, 1963]. For sketches as more general presentations of structured models, useful entry points are Barr and Wells [1999], Makkai and Paré [1989], and Adámek and Rosický [1994]. Barr and Wells [1992] is especially valuable because it identifies what ordinary sketches cannot present; it motivates the book's repeated care about enriched, higher, fibrational, and homotopical extensions.

Categorical database theory supplies concrete examples of the same declarative stance. Functorial data migration makes schemas and instances compositional [Spivak, 2012], while lifting problems express queries and constraints [Spivak, 2014]. Readers coming from logic may also consult Mac Lane and Moerdijk [1992] for the relation among sites, local models, and global semantics. LINCS reuses these presentational ideas but adds an obstruction object, tangent transport, a repair language, and an empirical admission decision.

4
Tangent Learning Sketches

The base obstruction says which structural promise has failed, but not whether the gap lies in a particular model or in the theory used to describe it. Its tangent lift asks how that failure changes under admissible infinitesimal perturbations. LINCS calls the resulting obstruction *infinitesimal non-compositionality*, or INC.

This chapter develops the model-level construction: the sketch is fixed while its realization varies. That restriction is methodologically important. Sensitivity within $\mathrm{Mod}(\mathbb{S}, \mathcal{C})$ can localize a defective map, representation, or interaction. A failure that persists across all registered model directions instead creates evidence that the available model class or sketch may be incomplete. It does not by itself license a new axiom. Theory augmentation is the higher-level repair problem developed in Chapter 22.

A tangent lift preserves the type of the declaration. It does not turn every available derivative into a legitimate learning signal.

4.1 Tangent structure

A tangent category, formally defined in Section 0.11, abstracts the behavior of tangent bundles without requiring every object to be a smooth manifold [Cockett and Cruttwell, 2014b, Rosický, 1984]. Its tangent endofunctor is

$$T : \mathcal{C} \longrightarrow \mathcal{C},$$

with structure maps $p, 0, +, \ell, c$ satisfying the fiber-power, additive, coherence, and vertical-universality axioms reviewed there.

For a smooth manifold M, TM contains pairs (x, v) of a point and a tangent direction. For a smooth map $f : M \to N$,

$$Tf(x, v) = (f(x), Df(x)[v]).$$

The categorical axioms preserve the compositional fact

$$T(g \circ f) = Tg \circ Tf.$$

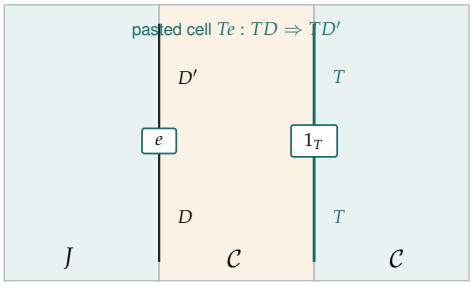
This is the elementary mechanism behind tangent factorization.

Definition 4.1 (Tangent model)**.** For a learning-sketch model $D : J \to \mathcal{C}$, its tangent model is

$$TD : J \longrightarrow \mathcal{C}, \qquad (TD)(j) = T(Dj), \quad (TD)(u) = T(Du).$$

The graphical calculus makes the formula $TD = T \circ D$ literal: the D-wire is pasted to the T-wire. If $e : D \Rightarrow D'$ is a coherent model repair, horizontal composition with the identity on T yields the tangent repair $Te : TD \Rightarrow TD'$. Thus differentiation transports a typed repair; it does not erase the naturality obligations that made the repair coherent.

(a) Tangent transport by horizontal pasting



(b) Tangent-structure glyphs

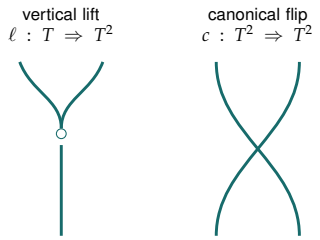


Figure 4.1: Tangent structure in the 2-categorical graphical language. The lift introduces another tangent level; the flip exchanges two tangent directions. The glyphs follow the calculus of Cockett and Cruttwell [2015].

4.2 *Tangent learning sketches*

Interpreting a graph in a tangent category is not yet enough. The declared learning constraints must remain meaningful after tangent lift.

Definition 4.2 (Tangent learning sketch)**.** A tangent learning sketch is a learning sketch $\mathbb{S} = (S, \mathcal{D}, \mathcal{L}, \mathcal{K})$ equipped with an action of tangent structure on its factorization problems. Whenever $D : J \to \mathcal{C}$ is admissible, TD is admissible and

$$\mathrm{Fact}_{\mathbb{S}}(D) \quad \longmapsto \quad \mathrm{Fact}_{\mathbb{S}}(TD)$$

is the declared tangent factorization problem.

This definition avoids an overly strong claim. The tangent functor need not preserve the quotient $J/\sim_{\mathcal{D}}$ as a colimit in a category of indexing shapes, nor every cone and cocone appearing in an arbitrary sketch. What is required is a lift on the particular class of factorization problems that the learning system declares.

Design principle

State tangent admissibility at the level of factorization problems. Do not infer it from the presence of automatic differentiation or from a blanket claim that tangent functors preserve every sketch construction.

One can package this requirement fibrationally. Let

$$p : \mathsf{Fact}(\mathcal{C}) \longrightarrow \mathsf{Learn}(\mathcal{C})$$

send a factorization witness to its underlying candidate model. Tangent stability asks for a lift T_{Fact} making

$$\begin{array}{ccc} \mathsf{Fact}(\mathcal{C}) & \xrightarrow{T_{\mathsf{Fact}}} & \mathsf{Fact}(\mathcal{C}) \\ {\scriptstyle p}\downarrow & & \downarrow{\scriptstyle p} \\ \mathsf{Learn}(\mathcal{C}) & \xrightarrow[T]{} & \mathsf{Learn}(\mathcal{C}) \end{array}$$

commute, so the fiber over D is transported to the fiber over TD.

4.3 *Infinitesimal non-compositionality*

Definition 4.3 (Infinitesimal non-compositionality)**.** The first-order infinitesimal non-compositionality of D is

$$\mathrm{INC}(D) := \mathcal{O}_{\mathsf{S}}(TD).$$

When iterated tangent structure is available,

$$\mathrm{INC}^{(n)}(D) := \mathcal{O}_{\mathsf{S}}(T^n D).$$

The base and tangent obstructions answer related but distinct questions. The object $\mathcal{O}_0(D)$ asks whether the realized diagram satisfies the declaration. The object $\mathcal{O}_1(D) = \mathrm{INC}(D)$ asks whether admissible infinitesimal perturbations satisfy its tangent lift.

Proposition 4.4 (Exact identities differentiate)**.** *Suppose two smooth composites $p, q : X \to Y$ agree after restriction to an open subset $U \subseteq X$. Then $T(p|_U) = T(q|_U)$ on TU. Consequently, an exactly commuting smooth learning diagram has trivial first-order route incompatibility along every admissible tangent direction based in that domain.*

Proof. Functoriality gives the tangent maps Tp and Tq. Since the base maps agree on an open domain, their derivatives agree there, hence $Tp = Tq$. □

The converse does not say that a small sampled tangent residual proves an exact base identity. It may reflect limited probes, low support, a null direction, or an insensitive observation.

4.4 *Why infinitesimal?*

The word *infinitesimal* describes the resolution of the diagnosis, not the permitted magnitude of the eventual repair. Tangent structure is

important to LINCS because it turns an undifferentiated failure into a family of typed directional questions. Given an admissible path of models D_t with initial direction $v = \dot{D}_0$, the tangent lift asks how each declared composition responds along v. Different directions can expose different arrows, interactions, or contexts even when a scalar loss assigns them the same value.

The construction has three further advantages. It is compositional because tangent transport respects composition. It is localizable because a direction can be supported on a declared component or overlap. And it makes null variation explicit: symmetry, reparameterization, or decision-irrelevant directions can be quotiented before repair effort is assigned. These properties make infinitesimal analysis a diagnostic microscope for structural failure.

When the relevant directions integrate, local repairs can assemble into a finite trajectory:

$$\dot{D}_t = v_t, \qquad D_0 = D, \qquad D_1 = D^+.$$

In such cases an apparently abrupt macroscopic change may be the accumulated effect of many locally intelligible changes. But tangent data alone neither guarantees that v_t integrates nor that D^+ lies in the same connected repair regime.

First-order analysis also has familiar blind spots. An obstruction varying as t^2 has zero first derivative at $t = 0$ without being locally constant. Higher Weil probes, jets, brackets, and curvature can expose some such effects. They still cannot manufacture a path to a disconnected model, a new primitive, or a new theory whose vocabulary is absent from $\mathbb{S}$.

Boundary

LINCS does not assume that every scientific or conceptual change is infinitesimal. Infinitesimal structure diagnoses the neighborhood of the current declaration. Finite integration, critical transitions, and discrete declaration changes require additional repair and admission mechanisms.

4.5 *The running equivariance example*

Return to the equivariance declaration from Chapter 3. Let X and Y be finite-dimensional normed vector spaces, and let the model $F_\theta : X \to Y$, input action $a_g : X \to X$, and output action $b_g : Y \to Y$ be smooth. The base representative is

$$E_{g,\theta}(x) = F_\theta(a_g x) - b_g F_\theta(x).$$

For an input perturbation v, the first tangent representative is

$$DE_{g,\theta}(x)[v] = DF_\theta(a_g x)[Da_g(x)v] - Db_g(F_\theta(x))\, DF_\theta(x)[v].$$

If a_g and b_g are linear, this reduces to

$$DF_\theta(a_g x)[a_g v] - b_g DF_\theta(x)[v].$$

The base audit asks whether the two finite routes agree. The tangent audit asks whether their local sensitivities intertwine the declared actions. A model can have a small base error at sampled points while its tangent error exposes a rapid departure nearby.

Example 4.5 (Parameter and input probes)**.** If parameters also move by $\dot{\theta}$, the full differential contains

$$\partial_\theta F_\theta(a_g x)[\dot{\theta}] - b_g \partial_\theta F_\theta(x)[\dot{\theta}].$$

Input probes v and parameter probes $\dot{\theta}$ answer different questions and should occupy different admission blocks.

4.6 *Probe semantics*

An automatic-differentiation system can produce many directional derivatives. LINCS requires the tangent site to say what those directions mean.

Definition 4.6 (Tangent site)**.** A tangent site for a realized learning sketch specifies:

1. the admissible base points or regimes;
2. the admissible tangent directions at each object;
3. how directions are transported along sketch arrows;
4. the cover over which tangent obstructions are observed; and
5. the null, gauge, or decision-invariant directions to be quotiented.

Examples include perturbations of tokens versus parameters, interventions versus domain shifts, policy directions versus advantage baselines, and semantic paraphrases versus warrant-changing argument edits. The derivative operator alone does not distinguish them.

Boundary

A parameter gradient is not automatically a semantic tangent direction. A map from parameter perturbations to deformations of the declared diagram must be specified and, when repair is claimed, shown to be sufficiently identified.

4.7 *Weil probes and higher order*

Tangent structure admits a useful algebraic classifier. In representable settings, an infinitesimal object supplies the dual-number probe; more generally, Weil algebras index tangent and higher-order prolongations [Leung, 2017a, MacAdam, 2022].

If W denotes the first-order dual-number object and

$$F : \mathsf{Weil}_1 \longrightarrow \mathrm{End}(\mathcal{C})$$

is the classifying functor, then $T = F(W)$. Iterated tangent structure probes mixed and higher infinitesimal behavior. This allows LINCS to distinguish:

- first-order route sensitivity;
- symmetric acceleration under a declared connection;
- antisymmetric order effects;
- curvature or holonomy; and
- higher jets of a factorization obstruction.

These refinements are optional structure, not part of every first-order INC object.

4.8 *Interactions and brackets*

Suppose a realized model generates admissible vector fields X and Y. Their Lie bracket $[X, Y]$ measures the antisymmetric discrepancy between the two local orders of flow. For a declared distribution $\mathcal{A} \subseteq TM$, a nonzero class of $[X, Y]$ in $TM/\mathcal{A}$ can indicate failure of that distribution to close. This is central to infinitesimal causality, BRIDGE/SKFM, ALLORA, and LASKO.

Yet a tangent category does not choose one universal interaction statistic. A connection ∇, when declared, also produces

$$A_\nabla(X, Y) = \frac{1}{2}(\nabla_X Y - \nabla_Y X), \qquad S_\nabla(X, Y) = \frac{1}{2}(\nabla_X Y + \nabla_Y X).$$

For a torsion-free connection, $2A_\nabla(X, Y) = [X, Y]$; the symmetric term depends on the connection and can carry different information.

Definition 4.7 (Interaction signature)**.** An interaction signature Ω_D is a declared family of operations on the admissible tangent distribution of D. It may include brackets, connection-dependent derivatives, curvature, canonical-flip comparisons, or higher jets.

Design principle

Choose an interaction signature because the application gives it meaning. Antisymmetry, curvature, and symmetric acceleration are different probes; none is a universal replacement for the tangent factorization obstruction.

4.9 *Natural LINCS: intrinsic tangent repair*

A tangent diagnosis identifies admissible directions in which an obstruction changes, but it does not yet say which corrective direction is smallest. A coordinate-dependent Euclidean norm on parameters answers that question only after choosing a presentation. *Natural LINCS* instead equips the model space with an intrinsic geometry and uses that geometry to select among typed repairs.

Let (M) be a smooth model object and let a typed obstruction be represented locally by

$$\mathcal{O} : M \longrightarrow E,$$

where $E \to M$ is an obstruction bundle. At $\theta \in M$, its linearized response is

$$D\mathcal{O}_\theta : T_\theta M \longrightarrow E_\theta.$$

Suppose that g is a declared Riemannian metric on model directions. A first-order natural repair is a solution of

$$\xi_\theta^\star \in \operatorname*{arg\,min}_{\xi \in T_\theta M} \frac{1}{2} g_\theta(\xi, \xi) \quad \text{subject to} \quad D\mathcal{O}_\theta[\xi] = -\mathcal{O}(\theta). \tag{4.1}$$

Thus the obstruction remains typed and potentially vector- or bundle-valued; the metric merely chooses the least intrinsic change that cancels it to first order. When exact cancellation is unavailable, a metric h on the obstruction fibers gives the regularized problem

$$\operatorname*{arg\,min}_{\xi \in T_\theta M} \frac{1}{2} \|D\mathcal{O}_\theta[\xi] + \mathcal{O}(\theta)\|_{h_\theta}^2 + \frac{\lambda}{2} \|\xi\|_{g_\theta}^2. \tag{4.2}$$

Definition 4.8 (Natural LINCS structure)**.** A Natural LINCS realization consists of a model object M, a typed obstruction object or bundle E, declared tangent and obstruction metrics g and h, a policy for quotienting null directions, and a retraction or integration map that returns an admitted tangent proposal to the model space. Its local repair rule is Equation (4.1), or an explicitly declared relaxation such as Equation (4.2).

For a scalar observation $L : M \to \mathbb{R}$, the metric induces the musical maps

$$g_\theta^\flat : T_\theta M \to T_\theta^* M, \qquad g_\theta^\sharp = (g_\theta^\flat)^{-1},$$

and the familiar natural-gradient vector is

$$\operatorname{grad}_g L(\theta) = g_\theta^\sharp(dL_\theta).$$

Natural gradient is therefore a scalar special case of the more general typed repair problem; it is not the definition of non-compositionality.

Design principle

The sketch declares what is broken; tangent diagnosis says how it changes; geometry chooses a presentation-invariant repair direction; admission decides whether to accept the resulting proposal.

Neither tangent-category axioms nor Cartesian closedness automatically supply a metric, a cotangent dual, an expectation operator, or an invertible Fisher matrix. These are additional semantic declarations. In an internal smooth setting, a metric may be represented by a bundle morphism $g^\flat : TM \to T^*M$, but its inverse may exist only after localization or quotienting. Singular directions are especially informative: they often mark parameter symmetries or observationally indistinguishable models. Natural LINCS therefore quotients such null directions before inversion, in the sense developed in Chapter 9, or uses an explicitly audited pseudoinverse or damping rule.

4.10 *From tangent diagnosis to learning*

When numerical observations are available, a computational realization may use

$$\mathcal{L}_{\text{base}}\big(\mathcal{O}_0(D_\theta)\big) + \lambda \mathcal{L}_{\tan}\big(\text{INC}(D_\theta)\big).$$

The categorical framework permits $\lambda = 0$. A nonzero tangent term is an empirical and semantic admission decision, not an axiom.

Admission contract

Before a tangent signal influences repair, the experiment must establish the semantics of its probes, quotient null directions, evaluate it on held-out support, and keep base, tangent, task, and safety outcomes separately visible.

The next chapter specializes tangent directions to smooth intervention protocols. This turns the abstract question "where does the factorization fail under variation?" into a causal one: do the declared intervention directions close under sequential local perturbation? A normal bracket component then marks a specific explanatory gap—a behavior generated by visible interventions but not expressible in their

declared span. Lie-bracket closure receives this causal interpretation only under additional statistical and structural assumptions.

Further reading

The axiomatic starting point is the theory of tangent categories developed by Cockett and Cruttwell [2014b]. Differential bundles and fibrations [Cockett and Cruttwell, 2018], and categorical connections [Cockett and Cruttwell, 2017], show how much differential geometry can be expressed without choosing coordinates. The graphical proof of the Jacobi identity by Cockett and Cruttwell [2015] shows how 2-categorical diagrams can turn long composites of tangent structure maps into local coherence rewrites. Leung [2017c] relates tangent structures to graph-presented Weil algebras and is the natural next source for the multi-infinitesimal constructions used later in the book.

For the geometry used to select intrinsic repair directions, see the original natural-gradient construction of Amari [1998] and the broader information-geometric treatment in Amari [2016]. These sources also clarify why a metric is additional statistical structure on a tangent model rather than a consequence of tangent structure alone.

For reverse transport, compare reverse derivative categories [Cockett et al., 2020] with reverse tangent categories [Cruttwell and Lemay, 2024]. The former abstracts reverse differentiation; the latter is closer to transporting cotangent information through an existing tangent structure. Standard differential geometry, including flows, brackets, distributions, and Frobenius integrability, can be reviewed in Lee [2012]. These sources help separate the generic tangent lift from the application-specific interaction signatures that LINCS calls INC.

5
Infinitesimal Causality

Chapter 3 supplied the categorical theory, and Chapter 4 supplied a way to discover where one of its models fails under variation. Infinitesimal causality now gives that transition a concrete scientific meaning. The tangent directions become causal only after we say how they are generated by a smooth intervention protocol on a statistical model. The protocol produces vector fields; their Lie brackets test whether the visible intervention directions close under sequential local perturbation.

In LINCS language, the protocol declares the tangent site, the visible span declares what the current model and causal vocabulary can express, and the normal bracket component is a typed obstruction to closure. It may diagnose a gap in the realized causal model, such as a missing visible direction. If the failure survives admissible model repair, it can instead motivate a theory-level proposal, such as adding a latent generator or enlarging the intervention language. The geometry locates the gap; independent evidence must still determine which explanation is admitted.

This chapter also separates two levels of intervention. Infinitesimal causality studies interventions on a modeled domain: a protocol changes a mechanism or regime and induces a tangent response. LINCS also permits interventions on the learning system that represents that domain: changing its visible span, estimator, causal vocabulary, cover, or model class. The first requires causal grounding. The second is a structural repair of the learner. A bracket residual can motivate either level, but cannot by itself decide between them.

The "Frobenius" in this chapter is the classical theorem on involutive distributions. It is distinct from the Frobenius algebras sometimes added to categorical models of classical data.

5.1 *A Three-Level Hierarchy*

The construction relates three mathematical levels.

1. A *Markov level* contains stochastic kernels together with the copy and discard operations of classical information [Fritz, 2020].

2. A *statistical level* contains a smooth family $\theta \mapsto P_\theta$, its scores, and the Fisher metric [Amari, 2016].

3. An *intervention level* contains specified local protocols and the vector fields they induce on the statistical manifold.

This hierarchy is distinct from Pearl's ladder of association, intervention, and counterfactual reasoning [Pearl and Mackenzie, 2018]. Pearl's hierarchy classifies the kinds of causal questions a model can answer. The hierarchy above instead separates the mathematical structures needed to give a local causal perturbation its semantics. The two classifications therefore cross-cut one another. A grounded intervention protocol can support Pearl's interventional level, but counterfactual claims require additional structure that relates outcomes across incompatible interventions; the three levels listed here do not supply that structure by themselves.

These levels communicate through an explicit realization. A tangent vector to a parameter manifold is not itself a stochastic kernel, and a derivative of a kernel is generally a signed map rather than a Markov kernel. Likewise, an arbitrary statistical perturbation is not automatically an intervention.

Design principle

The bridge from tangent geometry to causality requires a statistical model, a specified intervention protocol, and assumptions connecting that protocol to causal rather than merely distributional change.

This hierarchy also instantiates the declaration–realization distinction of Section 0.13. A causal vocabulary and its intervention protocols provide explicit structural commitments; learned statistical models and estimated vector fields realize them. Internal intuitionistic reasoning preserves unresolved or context-dependent causal claims, while bracket and integrability obstructions expose where the realization outruns the declared causal language. Neither the neural estimate nor the symbolic declaration is sufficient by itself.

5.2 *Statistical tangent semantics*

Let $\mathcal{M} = \{P_\theta : \theta \in \Theta\}$ be a regular q-dimensional statistical model. For

$$a = \sum_{k=1}^{q} a^k \partial_k \in T_\theta \mathcal{M},$$

the score representation is

$$s_a(x) = \sum_{k=1}^{q} a^k \partial_k \log p_\theta(x).$$

When the Fisher information is positive definite, the score map identifies $T_\theta\mathcal{M}$ with the finite-dimensional score span

$$\mathcal{T}_\theta = \operatorname{span}\{\partial_1 \log p_\theta, \dots, \partial_q \log p_\theta\} \subset L_0^2(P_\theta).$$

It does not identify the tangent space with all of $L_0^2(P_\theta)$.

A deterministic sufficient statistic $t : X \to S$ provides a useful link to the Markov description. Because t is deterministic, it respects copying:

$$\Delta_S t = (t \otimes t)\Delta_X.$$

For an exponential family,

$$p_\theta(x) = h(x) \exp\{\langle \theta, T(x) \rangle - A(\theta)\},$$

the sufficient statistic also generates the score directions,

$$\partial_i \log p_\theta(x) = T_i(x) - \mathbb{E}_\theta[T_i(X)].$$

Thus copyable summaries and statistical tangent directions are related, but they remain differently typed objects.

Example 5.1 (Gaussian location)**.** For $P_\mu = \mathcal{N}(\mu, \Sigma)$ with fixed positive-definite Σ, the score of $a \in T_\mu\mathbb{R}^d$ is

$$s_a(x) = a^\top \Sigma^{-1}(x - \mu),$$

and the Fisher metric is $g_F(a, b) = a^\top \Sigma^{-1} b$. Translation protocols generate constant coordinate fields, so all their brackets vanish.

Example 5.2 (Binomial family)**.** For $X \sim \text{Binomial}(n, p)$, the statistic X is sufficient and

$$\partial_p \log P_p(X) = \frac{X - np}{p(1-p)}.$$

This one-dimensional example has no nontrivial pairwise bracket test. It is a reminder that regular statistical tangent structure alone does not manufacture causal interactions.

5.3 *Smooth intervention protocols*

Pearl's interventions and potential-outcome estimands are usually formulated at finite scale [Pearl, 2009b, Rubin, 2005]. To differentiate an intervention, we need a smooth path that connects it to the observational model.

Definition 5.3 (Smooth intervention protocol)**.** A smooth local intervention protocol for mechanism i is a map

$$\Phi_i : (-\varepsilon, \varepsilon) \times U \longrightarrow \mathcal{M}, \qquad \Phi_i(0, p) = p,$$

on an open set $U \subseteq \mathcal{M}$. Its infinitesimal intervention field is

$$v_i(p) = \left.\frac{\partial}{\partial \alpha}\right|_{\alpha=0} \Phi_i(\alpha, p).$$

The strength α is not the assigned value in a hard intervention $\mathrm{do}(X_i = x)$. A hard-intervention family gives a smooth protocol only after we specify how it approaches the observational model. Different implementations aimed at the same substantive variable can therefore generate different fields.

Definition 5.4 (Visible intervention distribution)**.** For fields $v_1, \ldots, v_m$, define

$$\mathcal{D}_{\mathrm{vis}}(p) = \mathrm{span}\{v_1(p), \ldots, v_m(p)\} \subseteq T_p\mathcal{M}.$$

We work on a stratum where this distribution has constant rank r, with a local frame $w_1, \ldots, w_r$.

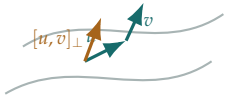


Figure 5.1: The normal bracket component records failure of the visible distribution to close.

When intervention protocols lift to coherent endotransformations of a realized model, the 2-categorical picture gives a useful local test. The two orders of intervention are vertical composites of the same two cells. Their first non-vanishing discrepancy is represented infinitesimally by the Lie bracket. The normal component is the part for which the current visible intervention language has no wire.

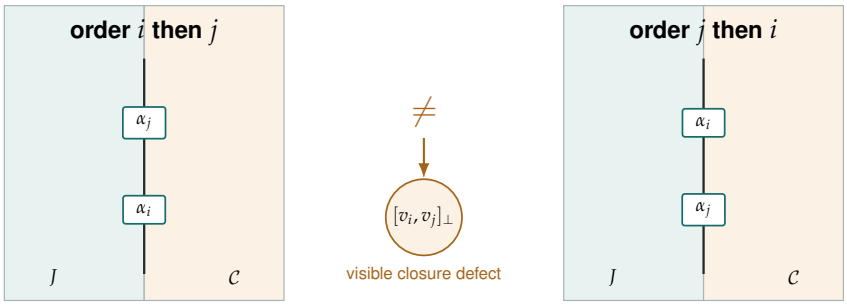


Figure 5.2: Sequential interventions as vertical composition. The picture is licensed only when the finite protocols have been typed as coherent model transformations. The bracket is the infinitesimal commutator of their generated fields, not a declaration that every intervention is a natural transformation.

5.4 *Bracket residuals and integrability*

Let g be a Riemannian metric on $\mathcal{M}$, and let $P_{\mathrm{vis}} : T\mathcal{M} \to \mathcal{D}_{\mathrm{vis}}$ be the orthogonal projection on the constant-rank stratum.

Definition 5.5 (Visible bracket residual)**.** For $u, v \in \Gamma(\mathcal{D}_{\mathrm{vis}})$, define

$$r(u, v) = (I - P_{\mathrm{vis}})[u, v] \in \Gamma(\mathcal{D}_{\mathrm{vis}}^{\perp}).$$

The projection makes the obstruction measurable. Its magnitude depends on the metric, but its vanishing does not.

Proposition 5.6 (Residual criterion)**.** *For a local frame $w_1, \dots, w_r$ of a constant-rank distribution, the following are equivalent:*

1. $r(w_a, w_b) = 0$ *for every* a, b*;*
2. $[w_a, w_b] \in \Gamma(\mathcal{D}_{\mathrm{vis}})$ *for every* a, b*;*
3. $\mathcal{D}_{\mathrm{vis}}$ *is involutive.*

Proof. The first two statements are equivalent by projection. If the frame brackets lie in the distribution, the Leibniz rule shows that the bracket of any two smooth linear combinations of the frame does also. The converse follows by applying involutivity to the frame sections. □

Corollary 5.7 (Frobenius integrability)**.** *On a constant-rank stratum, every visible bracket residual vanishes if and only if $\mathcal{D}_{\mathrm{vis}}$ is locally tangent to a foliation.*

Proof. Apply the classical Frobenius theorem [Lee, 2012] to the residual criterion. □

This is a closure theorem. It is not an acyclicity test, a causal-sufficiency criterion, or an identification theorem.

5.5 *Invariance and dependence*

If $F : \mathcal{M} \to \mathcal{N}$ is a diffeomorphism, naturality of the Lie bracket gives

$$[F_* u, F_* v] = F_* [u, v].$$

Consequently, involutivity and the zero-residual property are invariant when the fields, span, and metric are transported together. Replacing one smooth frame by another frame of the same distribution also preserves vanishing.

By contrast, the numerical residual norm, its tangential coefficients, and finite-sample thresholds depend on choices:

- the intervention protocols and their scaling;
- the visible span and the stratum on which its rank is constant;
- the metric used to define normal projection; and
- the estimation and separation assumptions.

Definition 5.8 (Separated stratum)**.** A constant-rank stratum is γ-separated for a chosen metric, frame, and protocol normalization when the frame Gram matrix has smallest eigenvalue at least $\gamma > 0$, and each tested residual has norm either zero or at least γ.

Separation is an estimation margin, not a consequence of statistical regularity.

5.6 *Two counterexamples that fix the interpretation*

Example 5.9 (Residual without a latent variable)**.** On $\mathbb{R}^3$, let

$$u = \partial_x, \qquad v = \partial_y + x\partial_z.$$

Then $[u, v] = \partial_z$, which is not in $\operatorname{span}\{u, v\}$ near the origin. The visible distribution is not involutive, yet the example contains no hidden random variable. A nonzero residual can arise because the declared visible controls are not closed under order-sensitive composition.

Example 5.10 (Latent structure with commuting visible fields)**.** Suppose latent variation changes a distributional family, while the two available visible interventions act as independent translations $u = \partial_x$ and $v = \partial_y$. Then $[u, v] = 0$ despite the latent structure. A zero residual shows closure of the tested visible span; it does not show absence of hidden causes.

Boundary

A nonzero bracket residual is a typed failure of visible closure, not by itself a certificate of latent confounding. A zero residual certifies involutivity only for the declared protocol and stratum; it does not certify causal completeness.

5.7 *Linearized copy and discard*

The intervention fields above live on a parameter manifold. Copy and discard live on sample objects. To compare them, take a differentiable path of finite-state stochastic kernels

$$K_\alpha : \mathbb{R}^X \longrightarrow \mathbb{R}^Y, \qquad K_0 = K,$$

and define its signed derivative

$$D = \left.\frac{d}{d\alpha}\right|_{\alpha=0} K_\alpha.$$

Proposition 5.11 (Infinitesimal normalization)**.** *Every differentiable path of stochastic kernels satisfies*

$$!_Y \circ D = 0.$$

Proof. Differentiate the stochasticity identity $!_Y \circ K_\alpha = !_X$. □

If $X = Y$, $K_0 = \mathrm{id}$, and the path preserves copying to first order, differentiation of

$$\Delta_X K_\alpha = (K_\alpha \otimes K_\alpha)\Delta_X$$

gives the coderivation identity

$$\Delta_X D = (D \otimes \mathrm{id} + \mathrm{id} \otimes D)\Delta_X.$$

Definition 5.12 (Copy defect)**.** The first-order copy defect is

$$B_D = \Delta_X D - (D \otimes \mathrm{id} + \mathrm{id} \otimes D)\Delta_X.$$

Discard compatibility is automatic for a valid stochastic path. Copy compatibility is much stronger: in a Markov category, generic stochastic maps do not preserve copying. It should therefore be tested only where the application declares deterministic, copyable information.

5.8 *The three IC compatibility tests*

The infinitesimal-causality calculus contains three differently typed tests.

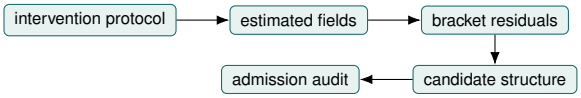
Test	Object tested	Vanishing means	Does not imply
Normalization or marginal test	signed kernel derivative	mass conservation or declared first-order marginal invariance	finite-scale causal irrelevance
Copy test	kernel coderivation defect B_D	first-order preservation of declared copyable information	generic stochastic compatibility
Bracket test	normal component of $[u, v]$	visible intervention distribution is involutive	absence or identification of latent causes

Table 5.1: The three tests share a compositional grammar but not a common type.

These tests resemble infinitesimal analogues of deletion, duplication, and reordering rules, but they are not a derivation of Pearl's do-calculus [Pearl, 2009b]. That stronger comparison would require a category of causal models carrying the relevant graph surgery, stochastic semantics, and a tangent construction compatible with both.

5.9 *From IC to discovery*

Infinitesimal causality supplies a population-level geometric diagnostic. BRIDGE and SKFM, developed in Chapter 11, turn that diagnostic into an estimation and model-selection pipeline. The distinction is important:



Each arrow requires assumptions. Field estimation needs support and regularity. Projection needs stable rank. Interpreting a residual as evidence about latent structure needs a causal model class and an identifiability argument. Selecting a repair needs held-out improvement and uncertainty control.

Admission contract

An IC signal may influence causal structure only when the protocol semantics, visible span, rank and separation conditions, estimation error, and competing non-latent explanations are explicit. Otherwise the correct output is the geometric obstruction itself, accompanied by abstention from a stronger causal claim.

This disciplined boundary is what makes infinitesimal causality useful to LINCS. It gives a precise interaction obstruction while keeping diagnosis, causal interpretation, and repair admission as separate stages.

It also illustrates how the causal lesson enters mainstream learning. The intervention protocol makes the external domain actionable; the LINCS sketch makes the model, estimator, and discovery pipeline actionable. A trustworthy system must record which level was changed. Improving the learner's structural model is not evidence that the corresponding domain intervention has been identified.

Further reading

The graphical and interventionist foundations are developed in Pearl [2009b] and Spirtes et al. [2000]; the potential-outcome perspective is represented by Rubin [2005] and Imbens and Rubin [2015]. These traditions clarify why an intervention protocol must be distinguished from an arbitrary distribution shift.

For categorical probability, Fritz [2020] develops Markov categories, Cho and Jacobs [2017] treats Bayesian inversion diagrammatically, and Fritz and Klingler [2023] gives a categorical account of d-separation.

String-diagram surgery [Jacobs et al., 2018], Universal Causality [Mahadevan, 2023], and intuitionistic j-do-calculus [Mahadevan, 2025c] explore alternative categorical semantics for causal reasoning. Infinitesimal Causality is complementary: it studies first-order response fields and their closure, and therefore requires additional identification assumptions before a geometric obstruction becomes a causal claim.

6
Infinitesimal Categorical Databases

TANGENT AND BRACKET-ENRICHED CSQL

Categorical databases turn a schema into a category and a database instance into a functor. This makes tables, foreign keys, path equations, queries, and data migration part of one compositional language [Spivak, 2012, 2014]. The original Democritus construction used this language to turn local causal graphs extracted from discourse into a queryable CSQL atlas [Mahadevan, 2025d]. Chapter 5 changes the local causal object. A smooth intervention protocol now produces vector fields, and Lie brackets test whether the visible intervention directions close.

This chapter asks what database object should retain that infinitesimal information. The answer has three layers. First, a database instance valued in a tangent category has a pointwise tangent lift. Second, a constrained database sketch admits that lift only when its declared constraints are tangent-stable. Third, Lie brackets require vector fields and a bracket operation in addition to the tangent functor. The resulting object is a *bracket-enriched infinitesimal database*: a queryable instance that stores generators, brackets, closure witnesses, quotient classes, and provenance without collapsing them into an ordinary graph.

The construction is not obtained by attaching derivatives to columns after the fact. The schema must declare which infinitesimal objects exist, how they migrate, and which bracket or closure relations are intended.

6.1 *From functorial databases to tangent instances*

Let $\mathcal{S}$ be a small category presenting a database schema and let $\mathcal{C}$ be a semantic category. The category of $\mathcal{C}$-valued instances is

$$\mathbf{Db}_{\mathcal{C}}(\mathcal{S}) = [\mathcal{S}, \mathcal{C}].$$

An object is a functor $I : \mathcal{S} \to \mathcal{C}$; a morphism is a natural transformation between instances. The familiar functorial database model takes $\mathcal{C} = \mathbf{Set}$. For infinitesimal data, however, the semantic category must carry nontrivial tangent structure.

Assume that

$$(\mathcal{C}, T, p, 0, +, \ell, c)$$

is a tangent category in the sense of Cockett and Cruttwell [Cockett and Cruttwell, 2014b]. Define an endofunctor on instances by first taking an instance morphism

$$\alpha : I \Rightarrow J$$

to be a natural transformation between two instance functors $I, J : \mathcal{S} \to \mathcal{C}$, and then setting

$$T_{\mathcal{S}} I = T \circ I, \qquad T_{\mathcal{S}} \alpha = T\alpha.$$

Here $T\alpha : T \circ I \Rightarrow T \circ J$ denotes postcomposition of the natural transformation by T; componentwise,

$$(T_{\mathcal{S}}\alpha)_s = T(\alpha_s) : T(I(s)) \longrightarrow T(J(s)).$$

Thus

$$(T_{\mathcal{S}} I)(s) = T(I(s)), \qquad (T_{\mathcal{S}} I)(f) = T(I(f)).$$

Proposition 6.1 (Pointwise tangent database)**.** *If $\mathcal{C}$ is a tangent category and $\mathcal{S}$ is small, the functor category $[\mathcal{S}, \mathcal{C}]$ inherits tangent structure pointwise. Its tangent functor is $T_{\mathcal{S}}$, and the structure maps $p, 0, +, \ell, c$ are defined componentwise.*

Proof. Limits in a functor category are computed pointwise whenever the corresponding limits exist in $\mathcal{C}$. For every object s of the schema, use the tangent structure map at $I(s)$. Naturality in s follows from naturality of the structure maps in $\mathcal{C}$. Every tangent category axiom then holds at each component because it holds in $\mathcal{C}$. The pullbacks selected by the tangent axioms are likewise computed componentwise. □

This proposition gives a precise version of the statement that a database category has a tangent functor. The tangent structure is inherited from the *semantic codomain*; it is not generated by a bare schema category.

Boundary

For **Set**-valued instances, the pointwise construction supplies no useful differential geometry unless **Set** is equipped with some additional nontrivial semantics. One must instead use smooth, synthetic, differential, or smoothly parameterized instance values. A finite table can remain as a discrete provenance layer while its numerical attributes live in a tangent category.

A useful hybrid semantics is a product category such as

$$\mathbf{Set}_{\mathrm{disc}} \times \mathbf{Smooth}, \qquad T(A, M) = (A, TM).$$

Keys, source identifiers, and provenance remain discrete; state spaces, parameters, fields, and residuals carry the nontrivial tangent structure.

More generally, one may work with bundles or smooth families of instances over a parameter object Θ.

6.2 *Instance lift, schema expansion, and constraint stability*

Three operations that sound similar must not be identified.

Pointwise instance lift. The schema $\mathcal{S}$ stays fixed and I is replaced by $T_{\mathcal{S}}I$. This is the construction proved above.

Syntactic tangent expansion. A new sketch $\tau\mathcal{S}$ is presented with tangent objects, projections, zero sections, lifts, or vector field symbols. This changes the database schema.

Internal tangent schema. If $\mathcal{S}$ itself has tangent structure, one may apply its tangent functor internally. A general database schema does not have such structure automatically.

The second construction is the one most useful for a persistent infinitesimal CSQL atlas. It makes the added infinitesimal columns and foreign keys explicit. It should be called a tangent *expansion*, rather than written $T\mathcal{S}$, unless a tangent structure on $\mathcal{S}$ has actually been supplied.

Database schemas often present more than paths. They may declare products, pullbacks, limits, equations, or lifting constraints. Let $\mathrm{Mod}(\mathbb{S}, \mathcal{C})$ denote the full subcategory of $[\mathcal{S}, \mathcal{C}]$ satisfying a database sketch $\mathbb{S}$.

Definition 6.2 (Tangent-stable database sketch)**.** A database sketch $\mathbb{S}$ is *tangent-stable in* $\mathcal{C}$ when $I \models \mathbb{S}$ implies $T_{\mathcal{S}}I \models \mathbb{S}$, with the declared comparison data preserved.

Path equations are preserved because T is a functor. A declared limit or pullback is preserved only when T preserves the relevant cone. The tangent category axioms guarantee selected pullbacks needed by tangent structure; they do not say that T preserves every database constraint.

Proposition 6.3 (Restriction to constrained instances)**.** *If* $\mathbb{S}$ *is tangent-stable, the pointwise tangent functor restricts to an endofunctor*

$$T_{\mathbb{S}} : \mathrm{Mod}(\mathbb{S}, \mathcal{C}) \longrightarrow \mathrm{Mod}(\mathbb{S}, \mathcal{C}).$$

If tangent stability fails, its missing comparison or failed cone is a typed obstruction rather than an infinitesimal database instance of the same declared type.

Proof. The first claim is the definition of tangent stability applied on objects; naturality of $T\alpha$ supplies its action on instance morphisms. For

the second, a lifted instance that violates a declared cone does not lie in the model subcategory. The failure is therefore a factorization or universal property obstruction of the kind introduced in Chapter 3. □

Design principle

Do not silently coerce $T_{\mathcal{S}}I$ back into the constrained database category. Either prove tangent stability, record the necessary comparison maps, or retain their failure as database-level obstruction data.

6.3 *Vector fields as natural database sections*

A tangent instance contains infinitesimal values but does not yet contain an infinitesimal intervention. A vector field on an instance I is a natural transformation

$$V : I \Longrightarrow T_{\mathcal{S}}I$$

satisfying

$$p_I \circ V = 1_I.$$

For every schema arrow $f : s \to t$, naturality requires

$$T(I(f)) \circ V_s = V_t \circ I(f).$$

Consequently, a database vector field is not an unrelated collection of per-table derivatives. Its components must respect the foreign keys and transformations declared by the schema.

Definition 6.4 (Infinitesimal database)**.** An infinitesimal database over $\mathbb{S}$ is a tuple

$$\mathfrak{D}_\partial = (\mathbb{S}, I, T_{\mathbb{S}}I, \mathcal{V}, \Lambda),$$

where I is an admissible instance, $T_{\mathbb{S}}I$ is its admitted tangent lift, $\mathcal{V}$ is a registered family of natural vector fields, and Λ records the observers and provenance by which their values are estimated.

If tangent fibers have negatives and an estimated field fails naturality, the route difference

$$\nu_f(V) = T(I(f))V_s - V_t I(f)$$

is a typed obstruction localized to the schema arrow f. Without negatives, the ordered pair of routes and its failed equality play the same role; no subtraction is implied. This distinguishes two failures that an ordinary edge table would conflate: incompatibility of a field with data migration, and non-closure among fields that are already compatible.

6.4 *Why the tangent functor does not supply a Lie bracket*

The tangent functor provides tangent objects and their structural maps. It does not, in every tangent category, choose a Lie bracket for arbitrary families of fields. We therefore make the needed enrichment explicit.

Definition 6.5 (Bracket-admissible semantics)**.** A tangent semantic category is *bracket-admissible* for a class of instances when it carries a bracket operation on the registered vector fields, and when related fields have related brackets. Concretely, if V_s, V_t and W_s, W_t are related by $I(f)$, then $[V_s, W_s]$ and $[V_t, W_t]$ are also related by $I(f)$.

Smooth manifolds satisfy this familiar related-fields property. Abstract tangent categories require appropriate additional hypotheses or an involution/Lie-algebroid enrichment [Burke and MacAdam, 2019]. The qualification matters: the bracket is an extra operation with laws, not an automatic synonym for tangency.

Proposition 6.6 (Pointwise bracket on database fields)**.** *In bracket-admissible semantics, the componentwise family*

$$[V, W]_s = [V_s, W_s]$$

is a natural vector field on I.

Proof. The registered fields V and W are natural, so their components are $I(f)$-related for every schema arrow. Bracket admissibility makes their componentwise brackets $I(f)$-related. This is exactly the naturality square for $[V, W]$; the section equation follows from the bracket construction on vector fields. □

Now choose a visible differential subbundle or distribution

$$j : \mathcal{D}_{\mathrm{vis}} \hookrightarrow T_{\mathcal{S}} I$$

generated by the registered intervention fields. A Lie bracket is an operation on *sections*, not a fiberwise bilinear map on tangent vectors. Assume therefore that the semantics supplies an object or sheaf of sections $\Gamma(\mathcal{D}_{\mathrm{vis}})$ and an object of database vector fields $\mathfrak{X}(I)$. The inclusion induces $\Gamma(j) : \Gamma(\mathcal{D}_{\mathrm{vis}}) \to \mathfrak{X}(I)$, and the bracket of included sections defines

$$\beta : \Gamma(\mathcal{D}_{\mathrm{vis}}) \times \Gamma(\mathcal{D}_{\mathrm{vis}}) \longrightarrow \mathfrak{X}(I).$$

Closure asks whether this operation factors through the visible section object:

$$\begin{array}{ccc} \Gamma(\mathcal{D}_{\mathrm{vis}}) \times \Gamma(\mathcal{D}_{\mathrm{vis}}) & \xrightarrow{\ \beta\ } & \mathfrak{X}(I) \\ & \overset{\tilde{\beta}}{\searrow} & \uparrow{\scriptstyle \Gamma(j)} \\ & & \Gamma(\mathcal{D}_{\mathrm{vis}}). \end{array}$$

Definition 6.7 (Bracket database obstruction)**.** The bracket obstruction $\mathcal{O}_{\mathrm{br}}(I, \mathcal{D}_{\mathrm{vis}})$ is the failure of β to factor through $\Gamma(j)$. If a normal quotient $q : \mathfrak{X}(I) \to \mathcal{N}$ of section objects is declared, it is observed by

$$\mathcal{O}_{\mathrm{br}} = q \circ \beta.$$

In Riemannian semantics this specializes to the projected normal bracket residual of Chapter 5.

This definition retains the non-scalar character of non-closure. A norm of $q\beta$ is one observer. The obstruction itself still records the context, field pair, normal type, and route by which closure failed.

6.5 *A bracket-enriched CSQL schema*

The construction can be presented through a tangent expansion $\tau\mathbb{S}$. The exact physical tables are application dependent, but the logical roles are stable.

Logical relation	Representative fields	Mathematical role	Typical query
Context	regime, source, population, chart, version	site over which an instance is interpreted	where was the field estimated?
Generator	field id, target, protocol, anchor, estimator	registered natural vector field	which intervention directions are visible?
Transport witness	schema arrow, source field, target field, residual	naturality of a database section	which fields fail under migration?
Bracket witness	ordered pair, context, coefficients, bracket value	antisymmetric interaction	which pairs interact by order?
Closure witness	visible span, quotient class, residual, uncertainty	factorization through $\mathcal{D}_{\mathrm{vis}}$	where does the span fail to close?
Evidence	data slice, probe, seed, estimator, confidence, provenance	observer and admission record	can the witness be reproduced?
Repair ticket	proposed generator, latent augmentation, invariant, status	typed change awaiting admission	which repair explains the obstruction?

Table 6.1: A logical schema for a bracket-enriched infinitesimal database. The relations store typed witnesses and provenance rather than only scalar bracket scores.

For a local frame $V_1, \ldots, V_r$, a bracket row may retain

$$[V_i, V_j] = \sum_{k=1}^{r} c_{ij}^{k} V_k + R_{ij},$$

together with the coefficient chart, quotient convention, uncertainty, and provenance of R_{ij}. The database should not treat the coefficients as globally comparable unless their frame-transport maps have been declared. The same geometric bracket can have different coordinates in different charts.

This yields CSQL-like questions that an ordinary causal graph cannot answer:

- Which intervention-field pairs fail closure in more than one regime?
- Does a closure witness survive quotienting of nuisance or gauge directions?
- Which schema migrations preserve a field but not its estimated bracket?
- Can one proposed latent generator repair several localized residuals?
- Does the repaired span close on held-out contexts without changing admitted causal effects?

6.6 *Queries and data migration at tangent level*

A schema functor $F : \mathcal{S} \to \mathcal{S}'$ induces the familiar restriction functor Δ_F by precomposition and, when they exist, left and right data migrations $\Sigma_F \dashv \Delta_F \dashv \Pi_F$ [Spivak, 2012]. Tangent lifting interacts differently with the three operations.

Restriction commutes strictly with the pointwise tangent lift:

$$T_{\mathcal{S}} \Delta_F = \Delta_F T_{\mathcal{S}'}.$$

For left and right migration, the canonical comparison maps have directions

$$\Sigma_F T_{\mathcal{S}} I \longrightarrow T_{\mathcal{S}'} \Sigma_F I, \qquad T_{\mathcal{S}'} \Pi_F I \longrightarrow \Pi_F T_{\mathcal{S}} I.$$

They are isomorphisms only when the tangent functor preserves the colimits or limits used to compute the corresponding Kan extensions.

Definition 6.8 (Query–tangent compatibility)**.** A data migration is query–tangent compatible on an instance when its canonical comparison map is an isomorphism, or when the declaration supplies an admitted weaker comparison sufficient for the intended query.

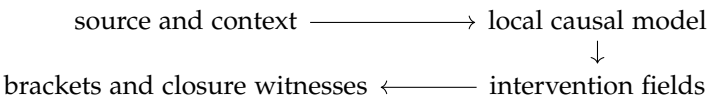
The failure of compatibility is another useful LINCS obstruction. It asks whether "query, then differentiate" agrees with "differentiate, then query." Unlike an ordinary numeric discrepancy, the failure identifies the schema migration and the universal construction responsible for the mismatch.

Design principle

Infinitesimal CSQL should expose whether a query commutes with tangent lifting. Foreign-key restriction does so pointwise; aggregation and constraint queries require explicit preservation hypotheses.

6.7 *From Democritus to BRIDGE and SKFM*

Democritus constructs local causal models from discourse and records their variables, claims, contexts, and provenance in a categorical atlas. Its local causal object is graph-like. Infinitesimal causality replaces a finite edge claim by a smooth response direction generated by a declared protocol. The infinitesimal database does not discard the earlier atlas; it expands it:



BRIDGE and SKFM, developed in Chapter 11, currently use bracket non-closure to prioritize latent-confounded causal discovery [Mahadevan, 2026b]. The database layer adds persistence and compositional memory. It can retain bracket witnesses across sources and regimes, compare alternative visible spans, record which extraction or estimation pipeline produced each field, and determine whether one repair explains several local failures.

The division of labor is therefore:

Infinitesimal causality supplies the intervention semantics and closure criterion.

BRIDGE/SKFM estimates fields and uses bracket geometry for causal search.

Database layer organizes the resulting objects, comparisons, queries, provenance, and repair history.

None of the three layers certifies causal identification by itself. A bracket row says how registered fields interact under a declared model and protocol; its causal interpretation still depends on the grounding and identification assumptions of Chapter 5.

6.8 The six-stage database workflow

Stage	Infinitesimal database action	Failure retained
Declare	choose schema, semantic tangent category, constraints, registered fields, quotient, and query language	underspecified tangent or causal type
Differentiate	construct $T_{\mathcal{S}}I$, fields, transport witnesses, and brackets	tangent, naturality, or bracket residual
Quotient	remove gauge, frame, nuisance, and decision-null directions	normal obstruction class with quotient provenance
Localize	index the witness by context, source, table, migration, field pair, and chart	cover on which the failure persists
Repair	add a generator, revise a protocol, refine a schema, transport a frame, or change an estimator	typed repair ticket and invariants
Admit	test held-out regimes, intervention semantics, migration stability, and finite effects	acceptance, rejection, quarantine, or abstention

6.9 Appearances, variations, and abductive repair

McCarthy's distinction between appearance and reality asks how an intelligent system can move from partial observations to stable objects and mechanisms that need not be present in the observational vocabulary [McCarthy, 2007]. A base database instance I records one organized appearance: observations, extracted claims, an admitted model, and their provenance. Its tangent lift $T_{\mathcal{S}}I$ enlarges that record with possible first-order variations. Registered vector fields make selected variations persistent and queryable; brackets record how pairs of such variations interact.

This enlargement is important, but it is not yet a database of counterfactuals. A tangent vector is a possible local variation in the chosen semantics. It acquires an interventional or counterfactual reading only when it is tied to a protocol, a mechanism replacement, and an identification contract. Under those conditions, an infinitesimal database can retain both what was observed and how a declared family of nearby interventions would change it. In that qualified sense it supplies part of the data infrastructure for a synthetic laboratory.

The distinction also sharpens the experience–jump–axiom cycle proposed by Zahavy [2026]. The jump should not be identified with the tangent lift itself. Tangent and bracket data diagnose the limits of the current declaration; the abductive step changes that declaration. A database-level reading is:

Experience. An admitted instance $I : \mathcal{S} \to \mathcal{C}$ organizes observations, contexts, and provenance under the current schema.

Diagnosis. Tangent instances, registered fields, and bracket residuals

locate variations that fail to satisfy an equation, close in a visible span, or commute with a query.

Jump. A typed repair $j : \mathcal{S} \to \mathcal{S}^+$ adds or revises an object, generator, path, constraint, protocol, or latent mechanism.

Axiomatization and admission. Constraints on $\mathcal{S}^+$ state the enlarged theory. Old evidence is transported, new consequences are computed, and the repair is admitted only after its finite and held-out implications survive the declared tests.

Thus the database does more than store a succession of models. It retains the typed obstruction that motivated a revision, the map from the old schema to the new one, and the evidence transported across that map. The "jump" is a schema migration supported by infinitesimal evidence, not an untyped leap outside the representational language.

Kan extension gives a precise account of one part of this migration. Suppose an intervened sub-schema $\mathcal{A}$ is included by $k : \mathcal{A} \hookrightarrow \mathcal{S}^+$, and a locally specified replacement is represented by $Q : \mathcal{A} \to \mathcal{C}$. When it exists,

$$\mathrm{Lan}_k Q$$

is the universal colimit-based propagation of that local datum to the expanded schema. This construction does not manufacture the intervention or certify its causal meaning. It propagates a replacement already supplied with an admissible intervention semantics. Causal validity still depends on the protocol, consistency conditions, identification assumptions, and admission tests.

Finally, the need for bracket enrichment marks the difference between individual and interacting variations. The tangent functor supplies first-order directions; it does not by itself supply the Lie bracket needed to compare their order of action. Once bracket-admissible fields have been declared, non-closure can reveal that the current visible span is insufficient. Such a residual can motivate an abductive schema expansion, but it does not force one: misspecification, estimation error, rank change, and an incomplete protocol remain competing diagnoses.

6.10 Scope and mathematical boundaries

The claims of the construction are modular and observe the following boundaries.

1. A pointwise tangent lift is a theorem about the instance category; it is not a theorem that an arbitrary physical database is differentiable.

2. Tangent-stable path equations do not imply preservation of every limit, join, aggregation, or query.

3. A vector field becomes causal only through a grounded intervention protocol.

4. A Lie bracket requires additional bracket-admissible structure and is not produced by the tangent functor alone.

5. Bracket non-closure detects a gap relative to a visible span. It does not distinguish latent confounding from misspecification, estimation error, rank change, or an incomplete protocol.

6. Storing a residual does not make two charts comparable. Frame transport, quotients, uncertainty, and provenance belong in the data model.

These boundaries are productive rather than merely cautionary. Each becomes a declared database constraint and therefore a possible obstruction. The database does not just hold the output of infinitesimal learning; it becomes an actionable model of how that output was constructed, compared, repaired, and admitted.

Further reading

Functorial databases and the adjoint data-migration pattern begin with Spivak [2012]; query and constraint semantics via lifting problems are developed by Spivak [2014]. Tangent categories and their abstract tangent-bundle structure are introduced by Cockett and Cruttwell [2014b]. Involution algebroids provide one route toward Lie-algebroid structure in tangent categories [Burke and MacAdam, 2019]. For the causal semantics of intervention fields and bracket closure, see Mahadevan [2026f]; for the associated latent-confounded causal-discovery application, see Mahadevan [2026b]. The present chapter combines these lines to define the mathematical interface among tangent instances, bracket enrichment, and categorical queries.

7
Infinitesimal Decisions

THE TANGENT LAYER OF UNIVERSAL DECISION LEARNING

Infinitesimal causality begins with a grounded intervention protocol and asks how its local effects compose. Decision making begins with a different problem: partial information is available in some contexts, yet an agent must act coherently in contexts it has not directly encountered. Universal Decision Learning (UDL) formulates this passage from local information to global behavior through left and right Kan extensions [Mahadevan, 2026l].

This chapter develops the tangent abstraction of that construction. An infinitesimal decision is not a small action. It is the first-order change in a universally extended decision semantics under an admitted perturbation of its data, context, constraints, or model. The resulting calculus applies to planning, constrained optimization, games, causal decisions, and online learning as well as reinforcement learning. GIRL, developed later in Chapter 15, is the Bellman and policy specialization of this more general layer.

The word *decision* refers to the entire semantics that generates and tests admissible behavior, not merely to a terminal discrete action. Its tangent may expose changes in candidates, constraints, values, equilibria, or policies.

7.1 *Decision making as universal extension*

Let $\mathcal{O}$ be a category of observed or locally accessible contexts, $\mathcal{C}$ a larger category of contexts in which decisions must be made, and

$$J : \mathcal{O} \longrightarrow \mathcal{C}$$

the functor that relates the two. A local decision model is a functor

$$F : \mathcal{O} \longrightarrow \mathcal{E},$$

where $\mathcal{E}$ contains the relevant decision objects: values, distributions, feasible sets, policies, strategies, or enriched utilities.

The pointwise left Kan extension

$$L_F = \mathrm{Lan}_J F$$

aggregates the locally available ways of reaching, explaining, or generating a context:

$$L_F(c) \cong \operatorname*{colim}_{(Jo \to c) \in (J \downarrow c)} F(o).$$

It is the candidate-generating side of decision making: rollout, interpolation, search, accumulation, or forward propagation.

The right Kan extension

$$R_F = \mathrm{Ran}_J(J^* L_F), \qquad J^* L_F = L_F \circ J,$$

assembles what remains compatible with the restrictions, continuations, or tests visible from a context:

$$R_F(c) \cong \lim_{(c \to Jo) \in (c \downarrow J)} L_F(Jo).$$

It is the coherence side: feasibility, backward consistency, equilibrium, or fixed-point semantics. In the simplest UDL presentation,

$$\mathrm{UDL}_J(F) = \mathrm{Ran}_J\big(J^* \,\mathrm{Lan}_J F\big).$$

The universal comparisons are especially transparent in the graphical language for 2-categories introduced in Chapter 0. For the right-hand panel below, write $G = J^* L_F$. The left Kan unit changes the direct local wire F into the composite $(\mathrm{Lan}_J F)J$; the right Kan counit changes the composite $(\mathrm{Ran}_J G)J$ back into G. The two cells face opposite directions, which is the precise graphical content of their left–right duality.

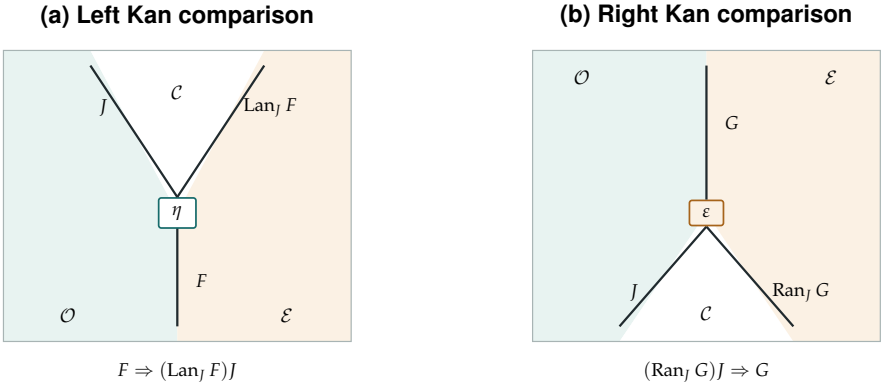
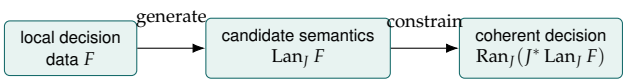


Figure 7.1: The unit and counit comparison cells for left and right Kan extension. The intermediate context category appears as a region created or removed at the comparison cell.



This formula is a semantic pattern, not a claim that every decision algorithm uses the same numerical implementation. The indexing functors, enrichment, order of operations, and approximation scheme are part of the declaration. The universal properties specify what a

correct extension must mediate; they do not select data, establish causal meaning, or guarantee that the required limits and colimits exist.

Design principle

A decision learner must expose both halves of its semantics: how candidates are generated and how global coherence is imposed. A final action can conceal failure in either half.

7.2 *A tangent site for decisions*

Let Θ be a space of admissible decision models and suppose the local data vary smoothly:

$$\overline{F} : \Theta \times \mathcal{O} \longrightarrow \mathcal{E}, \qquad F_\theta = \overline{F}(\theta, -).$$

The parameter θ may encode transition laws, preferences, payoff tables, constraints, forecasts, model parameters, or a mixture of these. For an admitted probe $v \in T_\theta\Theta$, write

$$\dot{F}_v = D_\theta \overline{F}(\theta, -)[v].$$

The induced global decision semantics is

$$U(\theta) = \mathrm{Ran}_J\big(J^* \,\mathrm{Lan}_J\, F_\theta\big).$$

Definition 7.1 (Infinitesimal decision)**.** When the displayed derivative exists, the *infinitesimal decision* in direction v is the typed pair

$$\big(U(\theta), \dot{U}_v\big), \qquad \dot{U}_v = D_\theta U(\theta)[v].$$

It records the base decision semantics together with its first-order response to the declared perturbation.

The definition does not identify v with an action. A direction can change the evidence used by the learner, its local model, a constraint, or the decision context itself. Calling the response a decision derivative is justified only after the probe language says which of these meanings is intended.

7.3 *Differentiating universal extension*

There are two conceivable routes from a local variation to a global one. One may first construct the global decision and differentiate it,

$$F_\theta \longmapsto U(\theta) \longmapsto \dot{U}_v,$$

or first differentiate the local data and then transport that variation through a registered tangent lift of the declared Kan construction,

$$F_\theta \longmapsto \dot{F}_v \longmapsto \mathrm{Ran}_J^T\big(J^* \,\mathrm{Lan}_J^T\, \dot{F}_v\big).$$

Probe	What varies	Tangent question	Additional semantics needed
Evidence probe	local observations or forecasts	how the decision changes with available evidence	sampling and observation model
Preference probe	reward, utility, or ordering	how tradeoffs change the selected behavior	elicitation and scale conventions
Constraint probe	feasible set or continuation test	which constraints become active	constraint qualification
Mechanism probe	transition or structural law	how behavior responds to a changed mechanism	causal or dynamical grounding
Context probe	state, history, information field, or game position	how nearby contexts alter the decision	geometry of the context space

Table 7.1: Infinitesimal decisions are typed by the probes that generate them.

These routes need not agree. Indeed, the second expression is meaningful only when the tangent objects and their base points inhabit a category in which the indicated lifted extensions exist. The superscript T records this extra structure; it is not obtained by applying an ordinary Kan extension to an untyped vector.

Write

$$\dot{L}_v^{\mathrm{dir}} = D_\theta(\mathrm{Lan}_J F_\theta)[v], \qquad \dot{L}_v^{\mathrm{ext}} = \mathrm{Lan}_J^T \dot{F}_v,$$

and define $\dot{U}_v^{\mathrm{dir}}$ and $\dot{U}_v^{\mathrm{ext}}$ analogously at the right Kan stage. Canonical comparison morphisms between the "extended" and "direct" tangent objects declare the desired compatibility; their existence and invertibility are not automatic.

Proposition 7.2 (Exact tangent transport)**.** *Suppose the comma-category shapes are fixed near θ, the relevant pointwise limits and colimits exist, and the chosen tangent construction preserves them. Then the canonical tangent comparison maps for the left and right Kan stages are isomorphisms. Consequently, tangent transport through the UDL composite is independent of whether differentiation is performed before or after universal extension.*

Proof. Pointwise Kan extensions are computed by the stated limits and colimits. Under the preservation hypotheses, applying the tangent construction to either pointwise universal object yields the universal object of the tangent-lifted diagram. The universal comparison maps are therefore isomorphisms at every context, and hence natural isomorphisms. □

The hypotheses matter. A maximum may change its active branch, a feasible set may change dimension, an equilibrium may bifurcate, or a comma category may change when the available information changes. At such points, ordinary derivatives can fail to exist even though a meaningful directional or set-valued response remains.

Boundary

Kan extension and differentiation do not commute by notation alone. Exact commutation is a theorem under preservation and regularity assumptions; its failure is itself a structural diagnostic.

7.4 *The infinitesimal decision obstruction*

Assume a computational realization supplies comparison morphisms

$$\chi_v^L : \dot{L}_v^{\mathrm{ext}} \longrightarrow \dot{L}_v^{\mathrm{dir}},$$

and

$$\chi_v^R : \dot{U}_v^{\mathrm{ext}} \longrightarrow \dot{U}_v^{\mathrm{dir}},$$

with directions adapted as necessary to the variance of the realization. LINCS retains their failures rather than immediately reducing them to a single loss.

Definition 7.3 (Infinitesimal decision obstruction)**.** Let $\mathcal{O}_{\mathrm{iso}}(\chi)$ denote the universal obstruction object for a sketch declaration that the comparison χ is invertible. The *infinitesimal decision obstruction* is the typed family

$$\mathcal{O}_{\mathrm{ID}}(\theta, v) = \big(\mathcal{O}_{\mathrm{iso}}(\chi_v^L), \mathcal{O}_{\mathrm{iso}}(\chi_v^R)\big).$$

Its first component localizes failure in candidate generation; its second localizes failure in consistency or selection.

In an additive numerical realization, observations of these components may be residuals

$$\Omega_v^L = \dot{L}_v^{\mathrm{dir}} - \dot{L}_v^{\mathrm{ext}},$$
$$\Omega_v^R = \dot{U}_v^{\mathrm{dir}} - \dot{U}_v^{\mathrm{ext}}.$$

The types of the two terms must first be aligned by the declared comparison. These residuals are computational witnesses, not the definition of the obstruction.

The separation matters. A rollout can vary smoothly while the final decision jumps because an active constraint changes. Conversely, the final action can remain fixed while the candidate landscape becomes unstable. The latter is invisible if one differentiates only the selected action.

7.5 Decision-relevant quotients

Many tangent changes do not affect the decision. Adding a common baseline to all action values, translating every feasible cost by the same constant, or reparameterizing an internal strategy may change coordinates while preserving behavior.

Let

$$\rho_\theta : U(\theta) \longrightarrow A_\theta$$

be the declared decision readout into an action, policy, ordering, feasible choice, or behavioral equivalence class. The decision-null directions are

$$N_\theta = \ker\big(T\rho_\theta : TU(\theta) \to TA_\theta\big),$$

whenever this kernel is defined. The decision tangent object is then modeled by the quotient

$$T_{\mathrm{dec}}U(\theta) = TU(\theta)/N_\theta.$$

Design principle

Differentiate the universal decision semantics, then quotient by variations that the declared decision rule cannot observe. Coordinate sensitivity is not yet decision sensitivity.

The quotient is application-dependent. In reinforcement learning it may remove action-common value shifts. In a game it may remove strategically equivalent mixed-strategy representations. In constrained optimization it may identify parameter changes that leave the active solution set invariant. No universal quotient can be chosen before the behavioral readout is stated.

7.6 Nonsmooth and set-valued decisions

Decision systems are often nonsmooth precisely where the interesting decision changes. An arg max can switch branches, a ReLU policy can cross an activation boundary, and a constrained optimum can acquire a new active constraint. Requiring a unique classical derivative at every such point would discard rather than analyze the event.

For a locally Lipschitz numerical realization, one may replace a derivative by a directional derivative, Clarke generalized Jacobian, or subdifferential. For a feasible-set valued decision, one may use tangent and normal cones. The infinitesimal decision then becomes set-valued:

$$\partial U(\theta)[v] \subseteq TU(\theta).$$

Nonuniqueness records competing local continuations of the decision.

Example 7.4 (A switching optimum)**.** Let

$$V_\theta = \max\{f_1(\theta), f_2(\theta)\}.$$

Away from a tie, the tangent follows the active branch. At $f_1(\theta) = f_2(\theta)$, the generalized derivative retains both branch gradients and their convex combinations. The tie is not numerical noise to be smoothed away automatically: it is a localized change in the candidate cocone and may alter the chosen behavior.

This is the decision-theoretic counterpart of the nonsmooth tangent language introduced in Chapter 1. Smooth infinitesimals and generalized derivatives serve the same structural purpose: they reveal how a declared composite can change locally without pretending that every local response is a single vector.

7.7 *The six-stage decision workflow*

Infinitesimal decisions instantiate the common LINCS workflow as follows.

Stage	Decision-level operation	Required record
Declare	specify J, F, the decision codomain, enrichment, Kan order, and behavioral readout	which contexts, candidates, constraints, and comparisons define success
Differentiate	probe evidence, preferences, constraints, mechanisms, or contexts	probe type, scale, support, and tangent semantics
Quotient	remove decision-null and representation-null variation	behavioral equivalence preserved by the quotient
Localize	separate rollout failure from consistency failure and assign it to a cover	candidate, constraint, continuation, or readout component implicated
Repair	change local data, context maps, enrichment, model, solver, or declaration	registered edit and invariants it must preserve
Admit	test finite decision quality and structural stability on independent contexts	utility, feasibility, safety, uncertainty, and transfer evidence

Table 7.2: The LINCS workflow specialized to universal decision semantics.

The repair may act at several levels. A parameter update changes F_θ inside a fixed declaration. A representation repair changes how contexts or decision objects are realized. A theory-level repair changes J, the enrichment of $\mathcal{E}$, or the family of admissible constraints. The obstruction can motivate any of these, but it does not choose among them without independent evidence.

Admission contract

An infinitesimal decision signal may guide repair only after its probe semantics, Kan comparison, decision quotient, nonsmooth regime, and localization are registered. Admission must then evaluate finite behavior: first-order coherence alone does not establish utility, feasibility, safety, equilibrium, or causal validity.

7.8 Decision modalities

The abstraction is useful because the same typed questions recur in decision problems that otherwise use different algorithms.

Planning. The left Kan stage aggregates partial paths, costs, or reachable outcomes. The right stage enforces compatibility with goals and continuation values. An infinitesimal decision records how changes in costs, dynamics, or a start context alter both the candidate path family and the globally coherent plan. A shortest-path switch is a nonsmooth change of active cocone.

Constrained optimization. Candidate generation supplies feasible proposals; consistency enforces constraints and optimality conditions. Tangent and normal cones distinguish directions along the feasible set from directions that violate it. Failure of regularity at an active-set change appears as a typed obstruction rather than an unexplained optimizer instability.

Games. Local strategies and payoff responses generate candidate profiles, while best-response compatibility selects equilibria. An equilibrium derivative describes the response to a payoff or information perturbation only while the equilibrium branch remains regular. Bifurcation or multiplicity requires a set-valued tangent semantics and an admission rule that states how a branch is selected.

Online decisions. History-indexed local losses are accumulated forward, while regret or feasibility comparators impose global tests. The tangent layer distinguishes sensitivity to the latest evidence from sensitivity to the comparator class itself. Drift can therefore be localized to the data stream, decision model, or admission contract.

Causal decisions. A mechanism-changing probe can transport a decision across an interventional context only when an intervention semantics has been supplied. The Kan construction organizes extension; it does not certify identification. Infinitesimal causality can provide

grounded local mechanism directions, while infinitesimal decisions studies how those directions change a downstream choice.

7.9 Relation to GIRL

GIRL is not a synonym for infinitesimal decision making. It chooses a particular decision modality and makes its structures concrete.

Table 7.3: GIRL is the reinforcement-learning realization of a broader infinitesimal decision layer.

Dimension	Infinitesimal Decisions	GIRL
Scope	planning, optimization, games, causal and online decisions, and RL	reinforcement learning
Base declaration	left/right Kan decision semantics	Bellman factorization for values and policies
Typical probe	evidence, preference, constraint, mechanism, or context direction	state, transition, reward, replay, or policy direction
Decision quotient	declared behavioral equivalence of the modality	action-common value variation and advantage contrasts
Consistency test	right Kan compatibility, feasibility, equilibrium, or fixed point	Bellman and policy consistency
Empirical realization	problem-specific and not fixed by the abstraction	GTD, Mirror–Prox, sampled transport, ESS and recovery gates

The distinction also clarifies the architecture of the book. UDL supplies the general base semantics; this chapter tangent-lifts that semantics; Deep LINCS compiles such typed diagrams into differentiable systems; and GIRL instantiates the resulting pattern for Bellman decision making.

Design lesson

Universal Decision Learning says that decision making extends partial behavior into new contexts by candidate generation and global consistency. Infinitesimal Decisions asks whether that extension remains coherent under typed local change. The answer is richer than a gradient of expected utility. It retains which half of the universal construction changed, which variations are behaviorally null, where nonsmooth switching occurred, and which finite decision tests must still be passed.

This is the general decision-theoretic role of tangent LINCS. It turns the local sensitivity of a decision system into an inspectable structural object without reducing all decisions to reinforcement learning or all failures to a scalar loss.

Further reading

Universal Decision Learning and its left/right Kan formulation are developed in Mahadevan [2026l]. Standard accounts of Kan extensions and their pointwise formulas are given by Mac Lane [1971] and Riehl [2017]. The categorical background in Chapter 0 explains the universal properties used here, while Chapters 4 and 5 supply the tangent and intervention semantics.

For the reinforcement-learning specialization, see Sutton and Barto [2018] and the GIRL development in Chapter 15. Pearl's intervention hierarchy [Pearl, 2009b] is essential for separating an ordinary parameter perturbation from a causally grounded mechanism change. The nonsmooth boundary connects to the generalized derivative and subdifferential discussion in Chapter 1; its central lesson is that a switching or set-valued decision requires a richer tangent object, not an arbitrary choice of one gradient.

8
Deep Learning in Non-Compositional Sketches

Deep Learning in Non-Compositional Sketches, or DLINCS, turns the structural language of the preceding chapters into a computational discipline for layered models. A deep model is not treated as one opaque parametrized function. It is presented as a typed learning sketch whose serial and parallel compositions are explicit, whose equations state the intended architecture, and whose tangent and reverse computations retain that structure.

We use *Deep LINCS* for the computational realization and DLINCS in formulas. Its role is to compile declared obstructions into trainable, auditable modules.

8.1 Why a deep specialization is needed

Chapter 3 described candidate models $D : J \to \mathcal{C}$. Chapter 4 lifted their factorization problems through T. A practical neural system adds three pieces of structure:

1. arrows carry parameters, often with sharing and gauge symmetries;
2. diagrams contain serial and parallel wiring at large scale; and
3. repair is computed through reverse information flow.

Ordinary backpropagation handles the third item brilliantly, but by itself it does not declare which paths, cones, or cocones should agree. Deep LINCS adds the sketch and then asks automatic differentiation to respect it.

Design principle

Architecture specifies the factorization problem; differentiation transports it; optimization proposes a repair. None of these three operations should silently redefine the other two.

This gives Deep LINCS a precise place in the symbolic–neural debate developed in Section 0.13. The sketch is not a symbolic reasoner attached to a network. It declares the typed compositional promises that the network realizes. The resulting obstruction is the semantic

interface between the declaration and the learned computation; reverse-mode differentiation then transports repair information through that same neural realization.

8.2 *Internal model spaces and parametrized maps*

The finite-product formalism of functorial backpropagation begins with a parametrized smooth map

$$f : P \times X \longrightarrow Y.$$

This is enough to compose parameterized modules: serial composition combines parameter spaces, and the monoidal product places modules in parallel [Fong et al., 2019]. It is not, however, enough for the semantics intended here. In the category of finite-dimensional smooth manifolds, the exponential object Y^X generally does not exist. Consequently, one cannot in general internalize the same family as a morphism

$$\widehat{f} : P \longrightarrow Y^X.$$

Deep LINCS therefore takes its primary ambient semantics to be a suitable smooth topos $\mathcal{E}$, following the Kock–Lawvere approach to synthetic differential geometry [Kock, 2006, Lawvere, 1980, Moerdijk and Reyes, 1991]. Cartesian closedness supplies the natural bijection

$$\mathcal{E}(P \times X, Y) \;\cong\; \mathcal{E}(P, Y^X),$$

so a parameterized implementation is equivalently an internal map into the space of models. The parameter object P need not equal all of Y^X; the transpose $\widehat{f}$ records which internal family of models is available. This is the same structural role played by internal function spaces in topos causal models.

The smooth topos and the tangent structure do different jobs. Cartesian closure supplies internal function spaces and currying. Infinitesimal structure supplies an object D and, on the appropriate infinitesimally linear or microlinear objects, a tangent construction represented by $TX = X^D$. A general tangent category need not be Cartesian closed; DLINCS requires these structures to coexist compatibly rather than deriving either one from the other.

Write $\mathbf{Para}_{\mathcal{E}}$ for the category whose parametrized arrows are pairs (P, f) internal to $\mathcal{E}$, modulo the chosen isomorphisms of parameter objects, and abbreviate it to **Para** below. The finite-dimensional Euclidean category of parametrized maps is a computational realization of this semantics, not its categorical foundation. A feed-forward network becomes a composite

$$X \xrightarrow{f_{\theta_1}} H_1 \xrightarrow{f_{\theta_2}} \cdots \xrightarrow{f_{\theta_L}} Y,$$

while branches, skip connections, adapters, critics, and local predictors are represented by a richer symmetric monoidal graph.

Definition 8.1 (Smooth learner)**.** A smooth learner from X to Y is a tuple

$$(P, I, U, r),$$

consisting of an implementation $I : P \times X \to Y$, an update

$$U : P \times X \times Y \longrightarrow P,$$

and a request map

$$r : P \times X \times Y \longrightarrow X.$$

The update changes local parameters; the request communicates desired change to the preceding interface.

Internally, the three displayed arrows are morphisms of $\mathcal{E}$. Cartesian closedness makes the implementation a map $P \to Y^X$, while a compatible reverse differential or reverse tangent structure supports the update and request operations. In the finite-dimensional Euclidean realization, a chosen loss and step size recover the functorial backpropagation construction of Fong et al. [2019]. That construction is an important operational model of DLINCS, but the internal smooth-topos semantics is strictly stronger than its finite-product baseline.

8.3 *Deep learning sketches*

Definition 8.2 (Deep learning sketch)**.** A deep learning sketch is a symmetric monoidal learning sketch $\mathbb{S}^{\otimes}$ together with a strong symmetric monoidal candidate model

$$D : \mathsf{FMon}(\Sigma) \longrightarrow \mathbf{Para},$$

where Σ is the typed generating signature and the assigned arrows are parametrized smooth modules.

The free symmetric monoidal category $\mathsf{FMon}(\Sigma)$ records both serial composition and parallel placement. Declared equations generate a quotient

$$q : \mathsf{FMon}(\Sigma) \longrightarrow \mathsf{FMon}(\Sigma)/\!\sim.$$

The candidate is compositional when it descends through this quotient and realizes any designated cones or cocones.

Definition 8.3 (Base DLINCS obstruction)**.** For parallel paths $p, q : x \to y$, let $\mathcal{O}_{\mathrm{eq}}(D(p), D(q))$ be the universal obstruction for the registered declaration that their images agree. Define

$$\mathcal{O}_0(D; p, q) = \mathcal{O}_{\mathrm{eq}}(D(p), D(q)).$$

For the whole sketch,

$$\mathcal{O}_0(D) = \mathcal{O}_{\mathbf{S}^{\otimes}}(D).$$

The symbol $\simeq$ is interpreted by the realization: equality of maps, an isomorphism, or an equivalence after quotienting a registered gauge. A supported statistical test may observe one of these declarations, but does not define the relation. A coordinate residual is likewise an observation of the obstruction, not its definition.

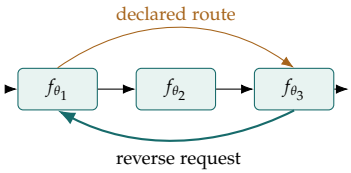


Figure 8.1: A deep sketch retains forward route obligations and reverse learner semantics.

8.4 Transport through backpropagation

Let $\mathcal{L}_{\varepsilon,e}$ denote the learner construction determined by a loss e and step size ε. If it is functorial and strong monoidal, then a satisfied architectural equation is transported:

$$D(p) = D(q) \quad \Longrightarrow \quad \mathcal{L}_{\varepsilon,e}D(p) = \mathcal{L}_{\varepsilon,e}D(q).$$

Proposition 8.4 (Backpropagation transports declared equations)**.** *Suppose D is a strong symmetric monoidal candidate and $\mathcal{L}_{\varepsilon,e} : \mathbf{Para} \to \mathbf{Learn}$ is a strong symmetric monoidal functor. If D factors through the equation quotient q, then $\mathcal{L}_{\varepsilon,e}D$ also factors through q.*

Proof. Equal formal paths have equal images under D. Functoriality preserves their equality under $\mathcal{L}_{\varepsilon,e}$, and strong monoidality preserves the declared parallel wiring. □

The converse need not hold, and equality of forward implementations need not force equality of arbitrary hand-designed update rules. The proposition depends on using the declared functorial learner construction.

Boundary

Forward agreement, tangent agreement, update agreement, and request agreement are four different audits. They coincide only under hypotheses supplied by the realization.

8.5 Tangent lifting of parametrized modules

For $f : P \times X \to Y$, ordinary tangent structure gives

$$Tf : TP \times TX \longrightarrow TY.$$

After the canonical product identification, this is again a parametrized map, now with parameter object TP. Hence tangent lifting descends to parametrized maps when it respects the chosen reparameterization equivalence.

Proposition 8.5 (Tangent descent). *If representatives $f : P \times X \to Y$ and $f' : P' \times X \to Y$ differ by a diffeomorphic reparameterization $\phi : P' \to P$, then their tangent representatives differ by $T\phi : TP' \to TP$. Therefore T defines a well-typed lift on the corresponding parametrized arrows.*

Proof. Differentiate $f' = f \circ (\phi \times \mathrm{id}_X)$ and use functoriality of T. □

Applying this lift to every arrow gives the tangent candidate

$$T_{\mathbf{Para}}D : \mathsf{FMon}(\Sigma) \longrightarrow \mathbf{Para}.$$

Its factorization obstruction is the first Deep-LINCS signal:

$$\mathcal{O}_1(D) = \mathcal{O}_{\mathrm{S}^\otimes}(T_{\mathbf{Para}}D).$$

8.6 Natural DLINCS updates

Chapter 4 introduced Natural LINCS as metric-aware typed repair. For a stochastic deep model with parameter object P and conditional law $p_\theta(y \mid x)$, a canonical candidate geometry is the Fisher metric

$$F_\theta(u, v) = \mathbb{E}_{(x,y)\sim\mu_\theta}\left[u[\log p_\theta(y \mid x)]\, v[\log p_\theta(y \mid x)]\right], \tag{8.1}$$

where $u, v \in T_\theta P$, and where the sampling law μ_θ is part of the declaration. For a scalar objective, the coordinate form of the resulting update is

$$\Delta\theta_{\mathrm{nat}} = -F_\theta^\dagger \nabla_\theta L.$$

More generally, Deep LINCS substitutes the Fisher metric into the constrained or regularized typed repair problems of Equations (4.1) and (4.2). The structural obstruction need not be collapsed into a single loss before the repair direction is selected.

The pullback Fisher metric of an overparametrized network is commonly singular: two parameter directions may induce the same observable model. From the LINCS perspective this is a quotient, not merely a numerical defect. The compiler should identify parameter gauges and observational null directions, solve on the quotient when possible, and record any practical approximation—pseudoinverse, damping, or structured curvature estimate—in the admission certificate.

This specialization also connects directly to reinforcement learning. On a policy manifold, the Fisher metric is weighted by a declared state-visitation law. Natural policy-gradient and natural actor–critic methods use that geometry to choose an intrinsic actor update. In a Natural-GIRL realization, the Bellman or intervention sketch still declares the repair target; policy geometry selects the actor direction, while the critic and compatible features provide an estimator. Geometry does not confer a causal meaning on the update or certify its admission.

Boundary

A smooth topos provides internal function spaces, and tangent structure provides admissible variation. Neither chooses the Fisher metric. Natural DLINCS requires a declared stochastic semantics, sampling law, quotient, and metric approximation in addition to the categorical architecture.

8.7 A four-level audit

For a declared relation $p \sim q$, Deep LINCS can retain four observations:

$$\begin{aligned} E_0 &= \omega_0\big(D(p), D(q)\big), \\ E_1 &= \omega_1\big(TD(p), TD(q)\big), \\ E_U &= \omega_U\big(U_p, U_q\big), \\ E_r &= \omega_r\big(r_p, r_q\big). \end{aligned}$$

The first measures finite forward disagreement. The second measures directional disagreement. The third compares parameter updates, and the fourth compares requests passed across module boundaries.

Level	Question	Typical computation	Common blind spot
Base	Do the declared forward routes agree?	output residual or factorization witness	nearby instability
Tangent	Do their admissible local variations agree?	JVP, VJP, or higher probe	wrong probe semantics
Update	Do the routes propose compatible parameter changes?	optimizer-state comparison	gauge and parameter sharing
Request	Do they send compatible credit backward?	interface cotangent comparison	agreement only at outputs

Table 8.1: Deep LINCS keeps the four levels visible rather than merging them into one training loss.

8.8 Weil probes beyond first order

In representable tangent categories, Weil algebras classify first and higher infinitesimal probes [Leung, 2017a,b]. A first-order dual-number probe yields the ordinary tangent lift. Products and iterates of Weil objects encode mixed directions and higher jets.

For a finite probe graph G, assign a first-order generator to each vertex and a mixed interaction probe to selected edges. The graph then

controls which directional interactions are evaluated without constructing the full dense higher derivative. This is valuable when a model exposes many candidate modules but the sketch declares only sparse interactions.

Design principle

Higher-order computation should follow the interaction signature. Evaluate only the mixed directions, brackets, or jets that correspond to declared structural questions.

8.9 *The DLINCS compiler*

The computational architecture can be organized as a compiler with a small interface.

Compiling a deep learning sketch

1: Parse typed objects, generators, serial wiring, and tensor wiring.
2: Construct the quotient relations and designated universal properties.
3: Instantiate every generator as a fixed or parametrized module.
4: Compile base obstruction observers for the declared relations.
5: Lift admitted objects and arrows to tangent or Weil probes.
6: Compile reverse-mode updates and requests compositionally.
7: Quotient parameter gauges and presentation-null directions.
8: If an intrinsic metric is declared, solve the typed tangent repair on the quotient and record any curvature approximation.
9: Retract the tangent proposal to parameter space without conflating it with admission.
10: Return separate base, tangent, update, and request certificates.

This compiler need not force every module to be trainable. Fixed transitions, incidence maps, restrictions, and aggregation operators are parametrized maps with terminal parameter objects. Trainable modules and fixed structure can therefore coexist in one sketch.

8.10 *Five recurring deep shapes*

The applications later in the book instantiate a small family of diagram shapes.

This classification does not prove the domain-specific claims of those systems. It shows that once their maps are realized as parametrized modules, the same serial, monoidal, tangent, and learner semantics can be reused.

Shape	Declaration	Deep realization	Later example
Parallel paths	two serial routes agree	shared modules with route-specific composition	GIRL, ALLORA
Restricted factorization	edge or local data arise from a smaller object	encoder plus constrained decoder	LINCS–RLHF
Descent diagram	local models agree on overlaps and glue	local networks and learned restrictions	SID
Typed apex	faces restrict from an information-rich object	view encoders, gauges, global decoder	RADAR
Controlled interaction	generated update fields close or compose	skills, adapters, intervention modules	LASKO; BRIDGE; SKFM

Table 8.2: Recurring deep-sketch shapes. Domain semantics determine what each factorization means.

8.11 Scaling and implementation choices

A naive all-path, all-direction audit is unnecessary. Practical Deep LINCS uses the locality already present in the sketch:

- generate a relation basis rather than enumerate every equivalent path;
- localize obstructions to the smallest subdiagrams that witness them;
- use Jacobian–vector and vector–Jacobian products instead of materialized Jacobians;
- sample sparse interaction probes from the declared signature;
- cache shared prefixes and suffixes of compared paths; and
- separate training-time observers from slower admission-time audits.

If there are R local relation generators and K admitted probes, the audit should scale with the evaluated local modules and approximately with RK, not with the number of all global path pairs. Chapter 9 develops the localization and admission mechanisms that make this possible.

Admission contract

A Deep-LINCS auxiliary signal is admitted only after its probe semantics, parameter quotient, computational budget, and held-out contribution are audited. The compiler must permit the tangent, update, or request coefficient to be zero.

8.12 What Deep LINCS adds

Deep LINCS does not propose a replacement for backpropagation. It gives backpropagation a typed structural specification. The sketch states what the system promises; tangent lifting exposes local instability; reverse composition transports credit through the same architecture; and admission decides which detected failures justify changing the model. This is the computational substrate shared by the application chapters.

Further reading

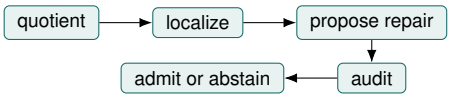
Categorical accounts of learning provide the closest conceptual background. Backpropagation as a functor appears in Fong et al. [2019]; reverse derivative categories and categorical gradient learning are developed in Cockett et al. [2020] and Cruttwell et al. [2022]. The categorical-deep-learning program [Gavranović et al., 2024] broadens this agenda from optimizers to algebraic descriptions of architectures and their constraints.

Natural gradient begins with Amari [1998]; Amari [2016] develops its information-geometric setting. For scalable deep networks, Kronecker-factored approximate curvature provides an influential structured approximation [Martens and Grosse, 2015]. Natural policy gradient [Kakade, 2001] and natural actor–critic [Peters and Schaal, 2008] show how the same geometric principle enters reinforcement learning. Natural DLINCS uses these methods to choose intrinsic repair directions while retaining typed obstruction and admission records.

On the machine-learning side, neural ordinary differential equations [Chen et al., 2018], normalizing flows [Papamakarios et al., 2021], diffusion models [Ho et al., 2020], and flow matching [Lipman et al., 2023] are useful examples of architectures whose semantics depend on paths, transports, or continuous dynamics. Deep LINCS does not subsume these models; it supplies a common way to declare which of their computational routes should compose, differentiate those promises, and decide whether a detected discrepancy should influence training.

9
Quotients, Localization, Repair, and Admission

An obstruction becomes actionable only after four further decisions. Which variation is real rather than presentational? Where does the failure live? Which corrections are allowed? What evidence licenses a change? LINCS assigns these questions to quotients, localization, repair languages, and admission contracts.



The stages form a control system around optimization. A gradient can propose a candidate, but it does not decide which quotient, site, or evidential threshold is legitimate.

9.1 Quotient before measuring

A learning system is usually presented with redundancy. Neural parameters have permutations and rescalings; low-rank adapters have factorization gauge; reward models have additive baselines; local trivializations have gauge transformations; causal fields depend on coordinates even when their distribution does not.

Definition 9.1 (Presentation quotient)**.** Let P be a presentation space and let $\pi : P \to M = P/\sim$ identify representatives with the same declared model behavior. A diagnostic $e : P \to Z$ descends to the quotient when there is $\bar{e} : M \to Z$ such that

$$e = \bar{e} \circ \pi.$$

If this condition fails, the diagnostic can change while the declared model does not. It is then a presentation artifact, not an obstruction on the model space.

Definition 9.2 (Projectable repair)**.** A field $V_P : P \to TP$ is projectable through π when there is a field $V_M : M \to TM$ satisfying

$$T\pi \circ V_P = V_M \circ \pi.$$

Alternatively, the realization may declare a gauge fixing or an equivariant horizontal lift.

Design principle

Measure and repair on the semantic quotient whenever possible. If computation must use representatives, declare the gauge and certify that the resulting signal descends.

9.2 *Localizing factorization failures*

Let $J = \mathrm{Path}(S)$ index a learning sketch and let $\mathcal{U} = \{J_i \hookrightarrow J\}_{i \in I}$ be a cover by subsketches. A global obstruction can be restricted to every J_i, producing local obstruction data.

Definition 9.3 (Obstruction localization)**.** A localization of $\mathcal{O}(D)$ over $\mathcal{U}$ consists of local objects

$$\mathcal{O}_i(D) = \mathcal{O}_{\mathsf{S}|_{J_i}}(D|J_i)$$

together with restriction maps on overlaps and any coherence needed to compare them.

Localization answers more than "which parameter has a large gradient?" It identifies declared subdiagrams in which a factorization, limit, colimit, or closure obligation fails. A claim of smallest support additionally requires a refinement order and a cover capable of detecting minimal witnesses.

Proposition 9.4 (Tangent stability of localization)**.** *Suppose T preserves the restrictions and overlap constructions used by $\mathcal{U}$, every restricted tangent model is admissible, and the obstruction assignment is natural under these restriction isomorphisms. Then tangent lifting commutes with localization:*

$$\mathrm{INC}(D)|_{J_i} \cong \mathcal{O}_{\mathsf{S}|_{J_i}}(T(D|J_i)).$$

Proof. By the preservation hypotheses, $(TD)|_{J_i} \cong T(D|_{J_i})$. The two sides therefore present the same restricted tangent factorization problem, and the obstruction assignment transports the isomorphism. □

The hypotheses matter. Automatic differentiation of a global program does not by itself show that a designated equalizer, pullback, or descent object is preserved by the chosen tangent realization.

9.3 *Compatibility is not effectivity*

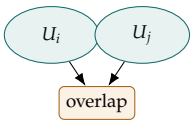
Local objects may agree on overlaps without arising from a global object in the declared model class.

Definition 9.5 (Compatible local family)**.** For local models D_i and restrictions to overlaps J_{ij}, compatibility means that the restricted models agree up to the declared coherence:

$$D_i|_{J_{ij}} \simeq D_j|_{J_{ij}}.$$

Definition 9.6 (Effective family)**.** A compatible family is effective when there exists an admissible global model D whose restrictions recover the D_i, with the required uniqueness or equivalence.

This separation is central to SID, RADAR, sheaf-structured argument systems, and distributed learning. A penalty on pairwise overlaps can reduce local disagreement while the compatible family remains outside the image of the global model class.



agreement need not produce a global lift

Figure 9.1: Compatibility is a local condition; effectivity is a global realizability condition.

Boundary

Localization diagnoses where a failure is witnessed. It does not guarantee that a local repair extends globally, that compatible repairs glue, or that the resulting global object lies in the admitted model class.

9.4 *The repair language*

A repair is not an arbitrary parameter step. It is a typed edit drawn from a declared language.

Definition 9.7 (Repair language)**.** For a model D, a repair language $\mathcal{R}(D)$ specifies admissible edits and their typing. It may include:

- parameter updates within a fixed architecture;
- changes to arrows, routes, or parameter sharing;
- additions or removals of objects, views, or constraints;
- modifications of covers, restrictions, or gluing maps;
- changes to an intervention, probe, or interaction signature; and
- abstention or escalation to a richer model class.

Repairs can therefore be *transverse*: they can change the presentation or admissible model class rather than moving only along the current parameter manifold. This is essential when the obstruction says that the current model cannot realize the required factorization.

Definition 9.8 (Repair proposal)**.** A proposal operator assigns candidates

$$\mathsf{Prop}_D : \mathcal{O}_{\mathsf{S}}(D) \rightsquigarrow \mathcal{R}(D).$$

The dashed arrow indicates that proposal can be set-valued, stochastic, or partial.

Gradient descent is one realization when the repair language consists only of parameter directions and a scalar observation is differentiable. It is not the general definition.

9.5 *Repair as structural surgery*

The repair language makes a learning sketch actionable in the same broad sense that a causal model is actionable. A causal intervention replaces a target mechanism while invoking a modularity assumption for the mechanisms not targeted. A LINCS repair similarly specifies an edit together with the obligations it promises to preserve.

Definition 9.9 (Structural intervention)**.** A structural intervention on an actionable learning model $\mathfrak{L} = (\mathbb{S}, D, \Lambda, \mathcal{R}, \mathcal{A})$ is an edit

$$e : (\mathbb{S}, D) \longrightarrow (\mathbb{S}_e, D_e)$$

equipped with:

1. a typed target in the realization or declaration;
2. an explicit class of non-target objects, arrows, equations, observers, and admission blocks to be preserved;
3. a propagation rule producing the candidate repaired behavior; and
4. a registered validation protocol, using evidence independent of the proposal signal, that must be evaluated before admission.

Four common cases form a hierarchy:

parameter	:	$D \mapsto D'$ inside a fixed realization class,
component	:	$f \mapsto f'$ for a selected module or morphism,
architecture	:	$\mathbb{S} \mapsto \mathbb{S}'$ with changed routes or covers,
declaration	:	$\mathbb{S} \hookrightarrow \mathbb{S}^+$ with new generators or axioms.

Each higher level enlarges what can be repaired and also what must be revalidated. Parameter repair can often be tested inside a fixed sketch. Declaration repair must transport prior evidence, preserve the successful fragment of the old theory, and face observations capable of distinguishing the augmented declaration from its predecessor.

The phrase *structural surgery* marks the methodological inheritance from causal inference, not an automatic causal interpretation. When the edited morphism denotes a causally grounded mechanism, the repair may realize an intervention in Pearl's sense. When it denotes an adapter, reward representation, warrant, or cover, it is an intervention

on the learning system. The categorical typing is shared; the causal semantics is not.

Boundary

A typed edit becomes a causal intervention only through an explicit mechanism model and realizable intervention protocol. LINCS supplies an actionable structural model and invariant-preservation obligations; it does not supply causal meaning merely by naming an edit "surgery."

9.6 *Functorial and descent-compatible repair*

Suppose $\alpha : D \to D'$ is a repair transformation. A tangent-compatible repair has a transported transformation

$$T\alpha : TD \longrightarrow TD'.$$

If D' factors through the declared quotient, then a repair advertised as compositionality-preserving must retain that factorization.

For a cover, global and local repairs must also interact with restriction:

$$\mathrm{res}_i \circ R_G \cong R_i \circ \mathrm{res}_i,$$

with cocycle coherence on overlaps.

Proposition 9.10 (Gluing compatible local repairs)**.** *Suppose the admissible models form a stack over the chosen cover, the local repairs R_iD_i agree on overlaps with coherent descent data, and each repaired local model remains admissible. Then the repaired family glues to a global admissible model, unique up to the declared equivalence.*

Proof. This is the effectivity condition in the stack semantics applied to the compatible repaired family. □

The proposition is deliberately conditional. Without effective descent, local repair compatibility is only a necessary condition for a global repair.

9.7 *Observation profiles before scalarization*

A single number often hides the information needed to choose among repairs. LINCS therefore permits a profile-valued observation

$$\Psi_S(D, e)$$

containing local defects, overlap effects, tangent behavior, task outcomes, uncertainty, computational cost, and downstream side effects of candidate e.

Definition 9.11 (Repair identifiability)**.** A profile identifies repairs, relative to the declared equivalence, when

$$\Psi_S(D, e) = \Psi_S(D, e') \quad \Longrightarrow \quad e \simeq e'.$$

When this implication fails, the evidence identifies only an equivalence class or moduli space of repairs. Choosing one representative may still be useful, but the choice requires an additional preference, cost, or regularity criterion.

9.8 *Admission is a separate decision problem*

Proposal and admission should be separated. The proposal mechanism searches the repair language; the admission mechanism decides whether available evidence warrants deployment.

Definition 9.12 (Admission rule)**.** Given validation evidence $\mathcal{V}$, an admission rule is a partial decision

$$\mathcal{A}_{\mathcal{V}} : (D, e, \Psi_S(D, e)) \longrightarrow \{\text{accept}, \text{reject}, \text{abstain}\}.$$

A useful rule is lexicographic. Hard invariants and safety constraints are tested first; only survivors are compared on task and structural outcomes.

A conservative admission procedure

1. Verify typing, quotient descent, and repair-language membership.
2. Reject candidates that violate hard structural or safety invariants.
3. Test the repaired base obstruction on held-out support.
4. Test admitted tangent and interaction blocks separately.
5. Evaluate task benefit, uncertainty, compute, and regression risk.
6. Prefer the least complex repair inside the supported equivalence class.
7. Accept only if the declared margin is met; otherwise reject or abstain.

Admission contract

Localization does not imply effectivity, an available tangent signal does not mandate an update, and defect reduction does not prove recovery of a privileged architecture. Admission must audit structural validity, semantic relevance, and empirical benefit separately.

9.9 *Validated scalarization*

After the obstruction profile and admission blocks are explicit, a numerical realization may use

$$\mathcal{L}_D^{\mathrm{val}} = \Phi_0\big(\mathcal{O}_0(D)\big) + \sum_{k\in\Omega_D} g_k(\mathcal{V})\lambda_k\Phi_k\big(\mathrm{INC}_k(D)\big),$$

where $g_k(\mathcal{V}) \geq 0$ is selected from validation evidence. The model class must permit $g_k = 0$. An auxiliary tangent signal that does not help, does not transport semantically, or cannot be estimated reliably should not be forced into the objective.

Block	Typical null	Evidence	Failure response	Leakage barrier
Base structure	declared factorization holds	held-out routes or overlaps	repair or enrich model	freeze declarations first
Tangent structure	admitted lift is coherent	held-out probes	drop signal or revise site	fit gates on validation only
Task behavior	no practical improvement	task metrics and slices	reject candidate	separate final test
Safety/invariants	no forbidden regression	hard checks and red teams	automatic rejection	never trade against mean gain

Table 9.1: Admission blocks remain separately inspectable even when a final decision combines them.

9.10 *Uncertainty and stochastic consistency*

Suppose an interaction profile is estimated from K minibatches by $\widehat{\Psi}_K$, with target population profile $\Psi_{\mathbb{E}}$. A useful certificate separates variance and bias:

$$\mathbb{E}\left\|\widehat{\Psi}_K - \Psi_{\mathbb{E}}\right\|^2 \leq \frac{C}{K} + b_K^2, \qquad b_K \to 0.$$

Same-minibatch and independently crossed directional estimates can converge to the same population profile while having different variance. The estimator design is therefore part of the admission contract.

Three outcomes should remain available:

- *accept* when the repair clears the structural and empirical margins;
- *reject* when it violates a hard condition or is clearly inferior;
- *abstain* when support, power, identifiability, or model capacity is insufficient.

Abstention is not a failure of learning. It is the correct result when the evidence does not license a model-changing claim.

9.11 *Certificates and provenance*

Every admitted repair should emit a compact certificate:

1. the sketch and version of its declarations;
2. the presentation quotient and tangent site;
3. the localized obstruction witnesses;
4. the repair-language element that was proposed;
5. the before-and-after profile on validation support;
6. uncertainty, invariants, and rejected alternatives; and
7. the admission decision with its thresholds.

The certificate is especially important when LINCS changes more than parameters. It allows a reviewer to distinguish a structural edit from an optimization step and to reconstruct why the system considered the change licensed.

9.12 *Iterated repair and coalgebraic stabilization*

Repair can be iterated. Let x_n denote the obstruction state after the n-th diagnose–repair cycle, and let

$$x_{n+1} = F_D(x_n)$$

be the realized INC update. This is first of all a discrete dynamical system. A coalgebraic presentation additionally packages an observable certificate, for example

$$c : M_D \longrightarrow O \times M_D, \qquad c(x) = \big(\mathrm{cert}(x), F_D(x)\big),$$

which is a coalgebra for the endofunctor $O \times -$. This packaging exposes the successive observations of a system that repeatedly unfolds and repairs its infinitesimal behavior [Rutten, 2000].

Theorem 9.13 (Contractive stabilization)**.** *Let (M_D, d_D) be complete and suppose $F_D : M_D \to M_D$ is ρ-contractive for $0 \le \rho < 1$. Then there is a unique fixed point x_*, and*

$$d_D(x_n, x_*) \le \frac{\rho^n}{1-\rho} d_D(x_1, x_0).$$

If approximate updates satisfy $d_D(\widetilde{x}_{n+1}, F_D\widetilde{x}_n) \le \varepsilon$, then

$$\limsup_n d_D(\widetilde{x}_n, x_*) \le \frac{\varepsilon}{1-\rho}.$$

Proof. The first claim is the Banach fixed-point argument. Summing the geometric bound on successive differences gives the displayed estimate. For approximate updates, apply contractivity recursively and sum the accumulated ε-terms. This is the metric-coinductive stabilization pattern [Kozen and Ruozzi, 2009]. □

Contractivity is an additional hypothesis, not a generic property of the tangent tower. Coalgebraic language organizes repeated diagnosis; it does not guarantee convergence of every repair loop.

9.13 *The reusable contract*

The application chapters use the same operational spine:

1. **Quotient** presentation-null variation.
2. **Localize** the typed base and tangent obstructions.
3. **Propose** edits from an explicit repair language.
4. **Check descent** and global effectivity.
5. **Observe** a profile before scalarizing it.
6. **Audit** uncertainty, task behavior, and invariants.
7. **Admit, reject, or abstain**, and record a certificate.

This is the point at which LINCS becomes a learning architecture rather than only a diagnostic vocabulary. The later systems differ in their sketches, probes, quotients, and repairs; they share this separation of structural failure from permission to change the model.

Chapter 1 stated the axiomatic contract near the beginning of Part I. The present chapter has now supplied its operational realization. Chapter 10 next assembles both views into a general architecture for auditable reasoning. Part II can then ask how the same rules specialize across causal discovery, neural architectures, adapters, skills, and policies.

Further reading

Sheaves and sites organize localization and descent [Mac Lane and Moerdijk, 1992]. Standard category-theory texts supply the universal-property view of quotients and presentation invariance; see Mac Lane [1998] and Riehl [2017]. Mirror descent and Mirror–Prox show how geometry can select a numerical repair without redefining its structural obligation [Beck and Teboulle, 2003, Juditsky et al., 2011].

Admission has close analogues in double/debiased machine learning and targeted learning, which seek valid inference beyond in-sample fit [Chernozhukov et al., 2018, van der Laan and Rose, 2011].

10
Reasoning by Structural Repair

THE COLT ARCHITECTURE

The preceding chapters developed LINCS as a theory of structural diagnosis and repair. The same machinery also defines a general architecture for reasoning. A reasoning system maintains a typed model of its commitments, probes that model, records failures of composition, proposes localized changes, and changes its maintained state only when independent evidence admits the proposal. We call an auditable sequence of these operations a *Chain of* LINCS *Thought*, or *CoLT*.

We write *CoLT*, with a lowercase "o," to distinguish the architecture from COLT, the Conference on Learning Theory.

CoLT is not a renaming of ordinary chain-of-thought prompting. A generated rationale is a sequence of linguistic tokens. A CoLT trace is a sequence of typed structural states and evidence-bearing transition records. It need not reveal, imitate, or claim access to a model's private internal computation. Its purpose is to expose what the maintained system declared, what was observed to fail, what was changed, and why that change was admitted.

Boundary

A plausible narrative is not yet an auditable reasoning trace. CoLT requires typed commitments, explicit obstruction witnesses, registered repairs, and admission evidence. Fluency may be an observation of a trace, but it is not its semantics.

10.1 Reasoning as the evolution of a maintained model

Conventional presentations often identify reasoning with a list of propositions

$$p_0, p_1, \ldots, p_n.$$

Such a list suppresses the structure that makes an inference meaningful: the types of its terms, the rule connecting one statement to another, the

scope of its evidence, the alternatives it ruled out, and the authority that licensed revision.

LINCS instead treats reasoning as controlled evolution of a maintained realization

$$D_0 \rightsquigarrow D_1 \rightsquigarrow \cdots \rightsquigarrow D_n.$$

The dashed arrows are not assumed to be logical implications or optimizer steps. Each is a governed transition with a diagnostic history. A state can retain unresolved obstructions; a rejected proposal can leave the state unchanged; and an abstention can request a new observation rather than force a conclusion.

Definition 10.1 (CoLT reasoning state)**.** A CoLT reasoning state is a tuple

$$\mathfrak{C}_k = (\mathbb{S}_k, D_k, \mathcal{T}_k, \pi_k, \mathcal{U}_k, \mathcal{R}_k, \mathcal{A}_k, \mathcal{E}_k),$$

where

- $\mathbb{S}_k$ is the current structural declaration and D_k its maintained realization;
- $\mathcal{T}_k$ is the admitted tangent site of meaningful probes;
- π_k is the quotient that removes declared null variation;
- $\mathcal{U}_k$ is the cover used to localize and glue evidence;
- $\mathcal{R}_k$ is the registered repair language;
- $\mathcal{A}_k$ is the admission contract; and
- $\mathcal{E}_k$ is an evidence and provenance registry.

The declaration can contain propositions, but it can also contain causal mechanisms, database relations, policies, argument roles, source bindings, or interfaces among agents. CoLT therefore applies to reasoning with models, not only to deduction in a fixed propositional language.

10.2 The anatomy of a CoLT transition

A transition begins with a probe and ends with a decision about the maintained state. It preserves the distinctions introduced by the six-stage LINCS workflow.

Definition 10.2 (CoLT transition record)**.** Given a state $\mathfrak{C}_k$, a CoLT transition record is

$$\tau_k = (\delta_k, o_k, \bar{o}_k, \lambda_k, r_k, \gamma_k, a_k),$$

with the following typed fields:

1. $\delta_k \in \mathcal{T}_k$ is an admitted probe;
2. $o_k = (o_k^0, o_k^1)$ records the observed base and tangent obstruction values;
3. $\bar{o}_k = \pi_k(o_k)$ is the obstruction after quotienting null variation;
4. λ_k localizes $\bar{o}_k$ over $\mathcal{U}_k$;
5. $r_k \in \mathcal{R}_k$ is a proposed typed repair, possibly the null proposal or a request for more evidence;
6. γ_k is a certificate containing provenance, preserved obligations, tests, uncertainty, and rejected alternatives; and
7. $a_k \in \{\mathsf{admit}, \mathsf{reject}, \mathsf{abstain}\}$ is the result of applying $\mathcal{A}_k$.

Only $a_k = \mathsf{admit}$ authorizes the maintained successor $\mathfrak{C}_{k+1}$. Rejection and abstention append records to the trace without silently changing D_k.

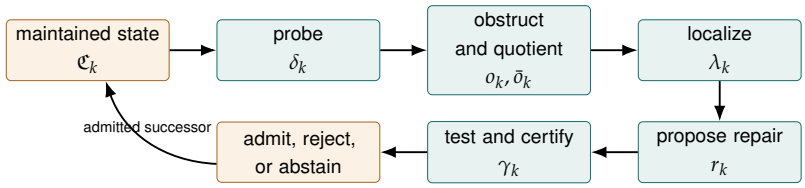


Figure 10.1: A CoLT transition. The return edge changes the maintained state only after admission. Rejection and abstention still extend the audit trace.

Definition 10.3 (CoLT trace)**.** A CoLT trace of length n is

$$\mathfrak{C}_0 \overset{\tau_0}{\Longrightarrow} \mathfrak{C}_1 \overset{\tau_1}{\Longrightarrow} \cdots \overset{\tau_{n-1}}{\Longrightarrow} \mathfrak{C}_n,$$

together with every rejected or abstained transition record encountered between the displayed admitted states. Each record must be type-correct for the state from which it was generated, and every state-changing arrow must carry an admission certificate.

The trace is consequently not just its successful path. Failed repairs, unresolved alternatives, and requests for new evidence are part of the reasoning object. They prevent the final conclusion from erasing how narrowly it was licensed.

10.3 *Constructive epistemic status*

Chapter 0 explained why internal excluded middle is not generally available in the smooth and sheaf semantics used by LINCS. CoLT turns that logical fact into an operational discipline. A system must not collapse lack of a witness into a witness of negation.

CoLT is therefore not a symbolic checker placed after a neural rationale. In the terminology of Section 0.13, its maintained sketch is a revisable declaration, a language or neural model may propose a realization or repair, and the public trace records the typed obstruction and the evidence that licenses a transition. Constructive status is what prevents the proposer from silently converting fluency into proof.

Table 10.1: CoLT preserves epistemic distinctions that a Boolean pass/fail label would collapse.

Status	Available evidence	Licensed action
Unobserved	No registered observer resolves the obligation	Acquire a probe or preserve uncertainty
Locally supported	A witness exists on one or more contexts	Retain with its scope; do not promote globally
Locally refuted	A typed counter-witness exists on a context	Localize the failure and block incompatible promotion
Compatible	Local witnesses agree on declared overlaps	Test effectivity in the admitted global model class
Globally effective	Compatible witnesses glue to an admitted realization	Submit the realization to independent admission
Repaired but untested	A registered edit removes the generating obstruction	Run preservation and held-out tests
Admitted	The admission contract accepts a certified successor	Update the maintained state
Abstained	Evidence, observability, power, or identifiability is insufficient	Keep the state and record the unresolved obligation

Holmes's elimination principle from Chapter 0 is valid when the alternatives are positively certified as exhaustive and the refutations are constructive. CoLT stores precisely those missing certificates. Without them, the last surviving hypothesis is a repair candidate rather than an admitted truth.

Design principle

A CoLT trace records the strongest judgment supported by the evidence. It never upgrades *not observed* to *false*, *locally coherent* to *globally effective*, or *repaired* to *admitted* without the corresponding witness.

10.4 Base, tangent, and transverse reasoning

At every state, CoLT retains two diagnostic layers:

$$O_k^0 = \mathcal{O}_{\mathsf{S}_k}(D_k), \qquad O_k^1 = \mathcal{O}_{\mathsf{S}_k}(TD_k).$$

The base obstruction asks whether the current commitments compose. The tangent obstruction asks whether that composition is stable under a declared family of nearby changes. A claim can therefore be coherent at the present state but fragile under paraphrase, evidence removal, population shift, mechanism perturbation, or reordering of interacting operations.

Infinitesimal diagnosis does not confine reasoning to infinitesimal change. When O_k^1 cannot be repaired inside the current realization, the repair language may propose a *transverse* change: a new object, path, axiom, cover, observer, or intervention vocabulary. The successor then has a changed declaration S_{k+1}. Old evidence must be transported to the new sketch, and the expanded theory must pass a new admission contract.

This is how CoLT accommodates an abductive jump without treating the jump as magic. Infinitesimal probes diagnose the boundary of the current theory; a transverse proposal changes that theory; transport and admission determine whether the proposal becomes part of the maintained model.

Boundary

The tangent signal may motivate a new hypothesis, but it does not identify or validate that hypothesis. Discovery, transport, and admission remain separate steps even when the proposal is generated by a powerful language or world model.

10.5 Local reasoning and global effectivity

Reasoning is often distributed over documents, agents, experiments, model runs, or argumentative subproblems. Let $\mathcal{U}_k = \{U_i \to U\}$ be the declared cover. A local CoLT record on U_i can contain a realization $D_{k,i}$, an obstruction, and a repair proposal. Two records are compatible when their restrictions agree on overlaps.

Compatibility alone is not global reasoning. The local states must glue to a realization in the admitted global model class, and their proposed repairs must induce a coherent global edit. Otherwise a system can produce mutually plausible local explanations that cannot all be true in one maintained model.

This distinction also governs multi-agent reasoning. Agreement among agents is neither necessary nor sufficient for truth. CoLT asks

instead whether their typed evidence is compatible, whether it is effective, which disagreement survives the decision quotient, and what authority can admit a global change.

10.6 An audit record, not private chain of thought

Chain-of-thought prompting can improve task performance by eliciting intermediate natural-language steps [Wei et al., 2022]. Those steps are useful artifacts, but their causal faithfulness to a model's internal computation cannot be assumed. Generated explanations can omit or rationalize features that influenced an answer [Turpin et al., 2023].

CoLT avoids depending on that assumption. Its public record contains only artifacts that can be typed and checked:

- the maintained declaration and versioned realization;
- externally inspectable claims, evidence, sources, and tool results;
- obstruction witnesses and their observers;
- registered repair proposals and preserved invariants;
- validation results, uncertainty, and admission decisions; and
- hashes, timestamps, model versions, or other provenance needed to reconstruct the transition.

A model may generate candidate rationales to populate this record, but the record earns authority from verification and admission rather than from a claim that the rationale exposes hidden cognition.

10.6.1 Chain-of-Evidence as an admission architecture

ScientistOne provides a recent operational example of this distinction. Its Chain-of-Evidence (CoE) standard requires every research claim to trace through recorded supporting claims to a grounding source [Meng et al., 2026]. Citation, numerical, methodological, and conclusion claims are checked against literature records, evaluator outputs, code, and supporting claims. Its post-hoc audit separately tests score reproduction, specification compliance, reference validity, and method–code alignment.

From a CoLT perspective, CoE is a concrete base-level admission sketch. Its objects include claims, references, code, task specifications, and experiment logs; its arrows include *cites*, *implements*, *reports*, and *derives from*. A missing source, irreproducible score, or method–code mismatch is then a localized factorization failure rather than a vague defect in the prose. Repair can remove or qualify the claim, correct the implementation, or rerun the experiment before admission.

The comparison also marks a boundary. CoE deliberately covers claim types that current tools can verify, and some of its checks use language-model judgments. CoLT treats such checkers as registered observers: their outputs are evidence with stated semantics and uncertainty, not truth by definition. CoLT also retains rejected and abstained transitions, rather than requiring the final successful provenance path to stand for the whole reasoning record.

A tangent evidence chain asks not only whether a claim reaches an admissible source, but whether that support survives declared changes of seed, evaluator tolerance, data slice, ablation, or evidence source.

CoE therefore supplies an important base construction for verifiable autonomous research. LINCS adds the question of stability: whether the claim–evidence paths continue to compose under admissible perturbation, and which typed repair is licensed when they do not.

10.7 SCoLT: the Toulmin specialization

The Toulmin application provides the clearest specialization. The Sheaf Chain of Toulmin Thought, SCoTT, supplies the base argument sheaf: local tickets contain claims, grounds, warrants, backing, qualifiers, rebuttals, and source bindings, and descent asks whether they glue. We call the CoLT architecture specialized to this argument sketch *SCoLT*, the *Sheaf-theoretic Chain of* LINCS *Thought*:

$$\mathrm{SCoLT} = \mathrm{CoLT}(\mathbb{S}_{\mathrm{Arg}}, \mathcal{T}_{\mathrm{Arg}}, \mathcal{U}_{\mathrm{Arg}}, \mathcal{R}_{\mathrm{Arg}}, \mathcal{A}_{\mathrm{Arg}}).$$

SCoTT is therefore the base sheaf-theoretic construction inside SCoLT. SCoLT adds tangent stability, localization, governed repair, and admission.

Chapter 19 develops this specialization and its initial structural evaluation.

10.8 A compact SCoLT trace

Suppose a local ticket contains the claim "treatment X reduces outcome Y," grounds from one study, a warrant that transports the study estimate to a target population, and a qualifier restricting the claim to that population.

1. **Declare.** The argument sketch requires source naturality, warrant transport, qualifier preservation, and descent.
2. **Probe.** Replace the target population while holding the surface claim fixed.
3. **Diagnose.** The base ticket still parses, but the tangent route through the warrant disagrees with direct transport of the claim.
4. **Quotient and localize.** Paraphrase variation is null; the surviving obstruction localizes to the population bridge and qualifier.

Argument component	CoLT role	Typical tangent or repair question
Claim	Maintained proposition with scope	Does paraphrase or context change silently strengthen it?
Ground	Evidence-bearing local section	Does removing or replacing one ground expose an unsupported route?
Warrant	Typed bridge from ground to claim	Is the bridge stable, sourced, and applicable in this context?
Qualifier	Boundary of licensed generality	Must the claim be weakened after transport?
Rebuttal	Registered counter-condition	Was an exception discharged, preserved, or silently erased?
Source binding	Provenance morphism	Does restriction preserve the evidence that licensed the step?
Global argument	Effective glued realization	Do locally coherent tickets form one admissible argument?

Table 10.2: SCoLT instantiates the general CoLT transition with Toulmin-typed argument structure.

5. **Repair.** Propose weakening the claim, adding transport evidence, or abstaining from population-level promotion.

6. **Admit.** Independent evidence determines whether the new warrant is licensed. If not, the maintained trace records abstention rather than silently generalizing the study.

This trace is useful even if every candidate sentence was generated by a different model. Its coherence comes from typed artifacts and admission contracts, not from continuity of an invisible monologue.

10.9 Minimal implementation contract

A system should not be described as CoLT unless it records at least:

1. a versioned declaration and maintained realization;

2. a typed distinction between probes, observations, and obstruction witnesses;

3. the quotient and cover relative to which localization is claimed;

4. a registered repair language with explicit targets;

5. preserved obligations and provenance for every proposal;

6. accept, reject, and abstain as distinct outcomes; and
7. an independently checkable admission certificate for every state change.

The contract permits implementations in theorem provers, databases, agent workflows, argument graphs, causal models, differentiable programs, and hybrid systems. Their common object is not a token format but a governed structural trace.

Admission contract

A CoLT successor is admitted only when the proposed repair is well typed, removes or appropriately reclassifies the targeted obstruction, preserves registered non-target obligations, transports provenance, survives the declared local-to-global checks, and passes evidence not used merely to generate the proposal.

10.10 Failure modes and open obligations

CoLT does not solve reasoning by definition. It makes several failure modes auditable:

- an incomplete declaration can omit the obligation that should have failed;
- an insufficient tangent site can miss a fragile inference;
- an overly aggressive quotient can erase semantically material change;
- an incomplete cover can support false localization or false consensus;
- a repair generator can search an impoverished hypothesis language;
- an admission test can reuse proposal data and overstate validation; and
- a certificate can faithfully record an invalid causal or domain assumption.

These are not defects hidden behind a single accuracy score. They are typed open obligations that later applications can test separately.

Further reading

The Toulmin model of argument is developed by Toulmin [1958]; modern argument-mining perspectives include Habernal and Gurevych

[2017] and Gupta et al. [2024]. Sheaf semantics and the internal logic of local truth are treated by Mac Lane and Moerdijk [1992].

For generated reasoning traces, chain-of-thought prompting was introduced at scale by Wei et al. [2022]. The distinction between a plausible rationale and a faithful account of model computation is examined by Turpin et al. [2023]. CoLT takes a different route: it treats public, typed, independently checked artifacts as the unit of audit rather than assuming that a verbal trace is privileged access to hidden computation.

Part II

Applications

11
Geometric Causal Discovery

BRIDGE AND SKFM

BRIDGE turns the infinitesimal-causality diagnostic of Chapter 5 into a causal-discovery front end. Regime changes produce estimated response fields. Directed influence proposes candidate arrows, while Lie-bracket residuals record failures of visible closure. A downstream causal learner then searches only inside the retained family. Spectral Kernel Flow Matching (SKFM) amortizes the fields and asks how much of the graph can be extracted directly from their geometry.

BRIDGE stands for *Bracket Residuals for Interventional Discovery and Geometric Estimation*. Its strongest present role is candidate generation, not stand-alone causal identification.

11.1 *Three data regimes, three outputs*

Let $V = \{X_1, \ldots, X_d\}$ be observed variables and let L denote unobserved variables. The meaning of a geometric score depends first on how the regimes were produced.

Known targets. A reference sample and interventional samples are observed, with target set $I_e \subseteq V$ supplied for each regime. BRIDGE returns a candidate mask, field-calibration diagnostics, and closure scores.

Unknown targets. Regimes are labeled but their targets are unknown. The graph–target pair is generally identifiable only up to Ψ-Markov equivalence; the natural population output is a Ψ-partial ancestral graph (Ψ-PAG) and invariant target information [Jaber et al., 2020].

General domains. Environment labels need not correspond to interventions on a shared structural causal model. Without additional assumptions, the output is a stability or transport diagnostic.

These regimes cannot be pooled under one "causal discovery" claim. With latent variables, a partial or maximal ancestral graph (PAG or MAG) may be the proper estimand; a directed acyclic graph (DAG) selected over observed nodes is only a restricted visible approximation [Spirtes et al., 2000, Pearl, 2009a].

Boundary

A large bracket residual is not a bidirected edge. A selected visible DAG is not a latent projection. Unknown targets, domain shifts, and known interventions require different estimands and different admission claims.

11.2 *The BRIDGE learning sketch*

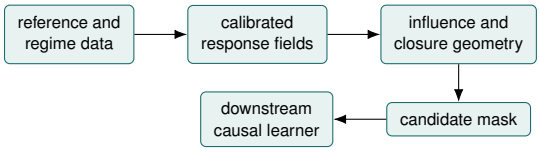
For a regime e with $P^{(e)} \ll P^{(0)}$, define the Radon–Nikodym ratio

$$\rho_e(z) = \frac{dP^{(e)}}{dP^{(0)}}(z), \qquad \mathbb{E}_{P^{(e)}} f = \mathbb{E}_{P^{(0)}}[\rho_e f].$$

This identity supplies a statistical interface. It does not identify a causal effect. A differentiable density-ratio or transport model converts the finite regime change into a response field v_e.

The sketch contains:

- observational and regime-specific probability objects;
- measure-change or transport arrows from the reference regime;
- coordinate observations of every response field;
- the visible distribution $\mathcal{D}_V = \operatorname{span}\{v_1, \ldots, v_q\}$; and
- a downstream graph or equivalence-class learner restricted by the geometric mask.



11.3 *Influence and closure are separate observers*

A population influence score for intervention i and coordinate j is

$$I_{ij} = \left(\mathbb{E}_{P^{(0)}}[(v_i)_j^2]\right)^{1/2}.$$

Large I_{ij} proposes that regime i changes coordinate j. It does not distinguish direct from mediated propagation without further conditioning or scoring.

Let Π_V project onto the visible span. The pairwise closure residual is

$$r_{ij}(z) = (I - \Pi_V(z))[v_i, v_j](z), \qquad R_{ij}^2 = \mathbb{E}_{P^{(0)}} \|r_{ij}(Z)\|^2.$$

This is the BRIDGE INC observer: it tests whether the fitted response family closes locally. The influence and closure blocks are kept separate because one proposes arrows while the other qualifies the adequacy of the visible field model.

Proposition 11.1 (Geometry-screen retention)**.** *Suppose every arrow in a prespecified target set $E^\star$ lies among the top-k population influences into its endpoint with margin $\gamma > 0$. If the estimated fields converge uniformly in the required L^2 and C^1 senses, then the empirical screen retains every member of $E^\star$ with probability tending to one. Residual classes separated by a margin around a fixed threshold are also consistently separated.*

Proof. Uniform convergence of the fields gives uniform convergence of the finite family of influence scores. An error smaller than $\gamma/2$ cannot reverse the top-k ordering at a target arrow. Continuity of $(u, v) \mapsto [u, v]$ in the chosen C^1 norm, together with consistent projection, gives uniform convergence of the residual scores. The threshold claim follows from the separation margin. □

Corollary 11.2 (Conditional downstream consistency)**.** *If the screen retains the representation required by an estimand $\mathcal{G}^\star$, and the downstream learner is consistent for $\mathcal{G}^\star$ whenever that representation is available, then the two-stage procedure is consistent for $\mathcal{G}^\star$.*

The corollary deliberately inherits the estimand and assumptions of the downstream method. The screen supplies retention, not a new identification theorem.

11.4 Falsifying a latent interpretation

Latent common causes can contribute to nonclosure by removing directions needed to express visible brackets. But the same signal can arise from target mismatch, ordinary domain shift, weak overlap, omitted observed coordinates, insufficient adjustment, or field error. Before a residual is called latent-sensitive, BRIDGE audits:

1. direct observational–regime distribution gaps;
2. overlap, density-ratio stability, and effective sample size;
3. held-out field continuity and calibration;
4. random, permuted, or null tangent controls;
5. sensitivity to adding measured adjustment coordinates; and
6. matched baselines for the actual data regime.

Admission contract

The phrase "latent-sensitive" means only that latent structure remains among the explanations not rejected by the diagnostic battery. It does not mean that a hidden variable, its dimension, or its incident edges have been identified.

11.5 *Nonclosure as a discovery trigger*

The closure residual can play a second role beyond screening. When

$$(I - \Pi_V)[v_i, v_j] \neq 0,$$

the visible response distribution has failed to contain an interaction that its own fields generate. This is an empirical obstruction to the declared causal geometry. It creates a discovery question: what must change for the response family to become adequate?

The first generated enlargement is

$$\mathcal{D}_V^{(1)} = \mathcal{D}_V + \operatorname{span}\{[v_i, v_j]\}_{i,j}, \qquad Q^{(1)} = \mathcal{D}_V^{(1)} / \mathcal{D}_V.$$

A nontrivial quotient $Q^{(1)}$ records directions required for closure but missing from the fitted visible span. Iterating the construction gives the Lie closure

$$\operatorname{Lie}(\mathcal{D}_V) = \mathcal{D}_V + [\mathcal{D}_V, \mathcal{D}_V] + [\mathcal{D}_V, [\mathcal{D}_V, \mathcal{D}_V]] + \cdots .$$

This completion is a mathematical hypothesis about missing directions, not a semantic identification of hidden causes.

Several explanations can produce the same quotient. A latent variable may supply an omitted direction. So may a measured but excluded coordinate, history dependence, an incorrectly pooled regime, a changing intervention target, support failure, or field-estimation error. The discovery phase must therefore branch across candidate declaration changes rather than map every residual to a latent node. One branch enlarges the state space; another refines the cover of regimes; another changes the transport model; still another rejects the fitted fields.

This is a small, executable instance of the outer discovery loop developed in Chapter 22. Nonclosure proposes that the current declaration is incomplete. Controlled additions of variables, contexts, or memory then generate competing completions. Admission asks which completion contracts the held-out residual while preserving calibrated influences and producing new testable consequences.

Design principle

Treat Lie-bracket nonclosure as a question generator. Its rank and localization can constrain candidate completions, but causal meaning must come from interventions, measured additions, and independent tests.

11.6 SKFM: amortizing the geometry

Kernel BRIDGE can be expensive because local fields are repeatedly estimated. SKFM replaces them with a conditional flow-matching network

$$f_\Theta(x, t, i)$$

indexed by intervention i. Automatic differentiation computes pairwise brackets. Their projected residuals form a Gram matrix G; a leading eigenspace E summarizes a low-rank visible footprint.

Spectral Kernel Flow Matching

1: Fit amortized response fields to observational–regime transports.
2: Verify held-out continuity, overlap, and influence calibration.
3: Compute selected Lie brackets by automatic differentiation.
4: Project brackets normal to the visible field span.
5: Form the residual Gram matrix and estimate its stable spectral rank.
6: Produce an influence mask and residual-footprint certificate.
7: Either pass the mask to a causal learner or apply a scoped ordered extractor.

The spectral coordinates summarize structured nonclosure. They are not identified latent variables without additional factor-model assumptions. Likewise, the direct extractor uses a supplied or learned order and a soft triangular penalty; this is a modeling choice rather than a general equivalence between DAGs and solvable Lie algebras.

11.7 Repair and admission

BRIDGE repairs the *search problem*, not necessarily the data-generating system. Its principal repair is a smaller candidate family:

$$\mathcal{G}_{\text{all}} \quad \rightsquigarrow \quad \mathcal{G}(\mathcal{E}_{\text{cand}}).$$

Other repairs include recalibrating a field, widening the mask, adding an adjustment coordinate, changing the regime interpretation, or switching the downstream estimand from a DAG to a PAG.

Admission is lexicographic:

1. reject fields that fail overlap or held-out calibration;
2. require target-arrow retention on controlled cases;
3. compare geometry to simpler distribution-shift diagnostics;
4. run a regime-appropriate downstream learner;
5. report screen and final-graph metrics separately; and
6. abstain from causal specificity not supported by the design.

11.8 What the experiments establish

The ten-node study uses six independently generated nonlinear DAGs (seeds 21–26), each with three injected latent-confounded visible pairs. The geometric screen is evaluated by candidate-arrow recall; a local Gaussian scorer, using the Bayesian information criterion (BIC), then selects parent sets inside that mask. The generator's topological order is supplied to this final step, and the best row from a small fixed hyperparameter grid is reported for each seed. It is therefore a screen-and-parent-selection test, not an order-free identification result. The Sachs diagnostic uses 853 measurements of 11 proteins and a 17-arrow canonical reference graph; its ordered parent-set row likewise uses a reference order, while pooled greedy equivalence search (GES) is the order-free comparator.

Experiment: BRIDGE/SKFM nonlinear-DAG sweep

On the ten-node nonlinear random-DAG sweep, the selected graphs average 12.8 arrows, precision 0.834, recall 0.907, and directed $F_1 = 0.864$. Direct SKFM extraction is less stable on these harder graphs even when the learned fields are accurate; the order-to-adjacency step is the principal bottleneck. This retained failure establishes the current division of labor: use geometry for high-recall compression and a score-based, latent-aware, or differentiable method for global selection.

Boundary

The Sachs study is a calibration frontier, not a solved biological discovery benchmark. The order-assisted parent-set diagnostic and the order-free GES comparisons answer different questions and must not be merged into one score.

Table 11.1: The strongest evidence supports calibrated geometric screening followed by a conventional causal learner.

Study	Geometry result	Downstream result	Boundary
Three-node chain	asymmetric response scores recover direction	exact visible chain	indirect scores remain positive
Nonlinear latent fork	dominant residual contracts when the missing coordinate is revealed	fork interpretation recovered in control	not a general latent test
Ten-node nonlinear DAGs	calibrated masks retain mean recall 0.907	local BIC gives mean directed $F_1 = 0.864$	best row selected from a small fixed grid
Sachs signaling	44 of 110 arrows retained; 15 of 16 pooled-GES arrows preserved	ordered diagnostic precision 0.889, recall 0.471	field correlation only 0.587 on average

11.9 *Design lesson*

BRIDGE/SKFM demonstrates a general LINCS pattern: an obstruction need not directly name the missing structure to be useful. A calibrated, nonspecific geometric diagnostic can still reduce a combinatorial search dramatically, provided its retention guarantee, falsification battery, downstream estimand, and abstention boundary remain visible.

This application also displays the two modeling levels introduced in the foundations. The SCM or causal graph is a model of the biological or experimental domain; the BRIDGE/SKFM sketch is a model of the discovery system that estimates fields, screens relations, invokes a downstream learner, and admits or rejects the resulting structure. LINCS makes the second system actionable without pretending that changes to its screen are interventions on the biological system. Domain surgery and learning-system repair can interact, but their semantics and evidence remain distinct.

Further reading

The classical causal-discovery background is given by Spirtes et al. [2000]. Greedy equivalence search [Chickering, 2002], FCI-style discov-

ery with latent and selection variables [Colombo et al., 2012], and interventional equivalence classes [Hauser and Bühlmann, 2012] provide important reference points for what can be identified under different regimes. Soft interventions with unknown targets are treated by Jaber et al. [2020].

Differentiable structure learning includes NOTEARS [Zheng et al., 2018], gradient-based neural DAG learning [Lachapelle et al., 2020], and interventional extensions [Brouillard et al., 2020]. For latent confounding, compare non-Gaussian and higher-order approaches [Cai et al., 2023, Morinishi and Shimizu, 2025, Tramontano et al., 2025] with the recent RelFCI treatment of relational data [Negro et al., 2025]. The Sachs protein-signaling study [Sachs et al., 2005] remains a useful empirical touchstone, but its interventions, preprocessing, and target graph must be read carefully. BRIDGE/SKFM belongs beside these methods as a geometric screen and field-estimation layer, not as a replacement for their identification theory.

12
Repairing Kan-Extension Structure

LINCS–KET

Kan Extension Transformers provide a clean architectural test of LINCS. The model contains a direct route and an incidence-restricted Kan-extension route to a common representation. Their intended agreement is structural. A post-training repair may reduce the route obstruction, but it is admitted only while the shared predictive task remains intact.

This chapter uses KET as the architecture and the later registered LINCS studies as the evidence. The original KET experiments are not retroactively counted as LINCS experiments.

12.1 *The declared Kan diagram*

Let X be a sequence or local data object, H a hidden representation, and Y the prediction object. A direct module

$$d_\theta : X \longrightarrow H$$

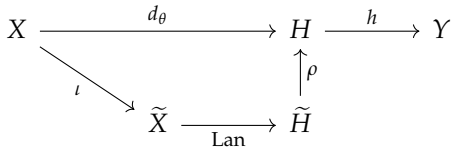
is compared with a structured route assembled from incidence transport, aggregation, and restriction:

$$k_\theta : X \xrightarrow{\iota} \widetilde{X} \xrightarrow{\mathrm{Lan}} \widetilde{H} \xrightarrow{\rho} H.$$

The learning sketch declares

$$d_\theta \sim \rho \ \mathrm{Lan}\ \iota.$$

Kan extensions characterize universal ways to extend data along a functor [Mac Lane, 1971, Riehl, 2017]. In the computational realization, the incidence route is finite and parametrized; LINCS audits its agreement with the direct route rather than claiming that every learned approximation satisfies an exact universal property.



This is an ordinary diagram of maps in the computational category. At the functorial level, the universal comparison that licenses the Lan route is a 2-cell of the form shown in Figure 7.1. The implemented finite route should not be promoted to such a cell by notation alone: naturality and the relevant universal property remain admission obligations. This is why LINCS observes the direct and structured routes instead of declaring them equal by design.

12.2 *Base and tangent obstructions*

At example x, the base route obstruction is

$$E_0(x) = d_\theta(x) - \rho\ \mathrm{Lan}\ \iota(x).$$

For an admitted input direction v, the tangent obstruction is

$$E_1(x; v) = Dd_\theta(x)[v] - D(\rho\ \mathrm{Lan}\ \iota)(x)[v].$$

The task block is evaluated after the shared decoder h. Thus route compatibility and prediction quality are observable separately.

Definition 12.1 (LINCS–KET profile)**.** For validation set $\mathcal{V}$, define

$$\Psi_{\mathrm{KET}}(\theta) = \big(L_{\mathrm{task}}, \|E_0\|_{\mathcal{V}}, \|E_1\|_{\mathcal{V},\mathcal{P}}, L_{\mathrm{shift}}\big),$$

where $\mathcal{P}$ is a fixed multi-probe tangent design and L_{shift} contains declared perturbation or length audits.

The fixed probes make before-and-after comparisons meaningful. Resampling one probe per candidate can turn admission noise into apparent improvement.

12.3 *The repair language*

The initial predictor is pretrained by the task loss. LINCS then freezes the shared decoder and most of the model, allowing only incidence and route adapters to change. A candidate repair is therefore a small, localized parameter edit:

$$\theta' = \theta + \Delta_{\mathrm{route}}.$$

It is evaluated against frozen validation data before being committed.

Blockwise LINCS–KET repair

1: Pretrain the KET model on the base prediction task.
2: Freeze the task semantics and the registered audit sets.
3: Propose a bounded edit to the incidence and route adapters.
4: Evaluate task, base-route, and multi-probe tangent blocks separately.
5: Admit only if structural blocks improve and task drift stays within cap.
6: Commit accepted edits; otherwise restore the previous state.
7: Report held-out task and structural audits without further selection.

12.4 *Why a joint norm failed*

The first registered experiment concatenated task, base, and tangent residuals into one normalized admission score. It preserved the task and reduced base incompatibility, but the held-out tangent block worsened. This is a five-seed synthetic architecture test on a causal edge-local sequence task. It uses the implemented incidence KET route, independently checks automatic-differentiation tangents by finite differences, and evaluates length and perturbation shifts after repair. Its purpose is to isolate the admission mechanism, not to establish language-model scaling.

Audit	Pretrained	Joint gate	Change	Reading
Task RMSE	0.06114	0.06129	+0.25%	predictive semantics preserved
Base Kan RMS	0.84982	0.78928	−7.12%	finite route agreement improved
Tangent Kan RMS	1.46421	1.61963	+10.61%	registered promise failed
Length-7 RMSE	0.12701	0.12515	−1.46%	no collapse on this shift

Table 12.1: The retained first result. Improvement of a joint norm did not imply improvement of every semantic block.

An equal-weight penalty comparator drove base and tangent residuals down to 0.0661 and 0.1654, but raised task RMSE from 0.0611 to 0.4477. It made the two routes agree by destroying their shared predictive meaning.

Boundary

Small compatibility residuals are not useful when both routes have collapsed to the same bad computation. The task block is not a soft preference; it is part of the semantic identity of the diagram.

12.5 *Blockwise admission*

The follow-up experiment was registered separately with fresh seeds, four fixed training and validation probes, pretrained block normalization, and lexicographic guards. It met all criteria.

Audit	Pretrained	Blockwise LINCS	Change	Result
Task RMSE	0.05629	0.05633	+0.06%	inside task guard
Base Kan RMS	0.94531	0.88325	−6.56%	improved
Tangent Kan RMS	1.87822	1.73857	−7.43%	improved in 5/5 seeds
Perturbed RMSE	0.05743	0.05730	−0.23%	preserved
Length-7 RMSE	0.10792	0.10758	−0.31%	preserved

Table 12.2: Fresh-seed blockwise multi-probe admission resolves the specific reliability failure of the first gate.

The result does not prove that blockwise admission always outperforms a joint score. It establishes a stronger contract: every admitted repair satisfies the declared task, base, tangent, and normalized-score guards.

12.6 *A small language-model test*

The Penn Treebank (PTB-small) study uses a word-level causal KET. Pretraining fits next-token prediction; LINCS then repairs only the incidence and route adapters. A deterministic 512-token vocabulary, context length eight, 512 training windows, 32 frozen repair-validation windows, and 128 test windows make the study deliberately small. The three seeds share five repair steps and four fixed train and validation tangent probes; embeddings, backbone, normalization, and vocabulary decoder remain frozen during repair.

Experiment: PTB-small registered result

Across three held-out seeds, base Kan RMS falls 2.35% and tangent Kan RMS falls 2.32%. Overall test cross-entropy changes by +0.11%, perplexity by +0.48%, and non-<unk> cross-entropy by −0.41%. Every seed improves both structural blocks, so the registered criterion passes.

The vocabulary is truncated to 512 tokens and the test target <unk> rate is 32.62%. The reported perplexity is therefore not comparable to standard full-vocabulary PTB. The experiment supports post-training structural repair with nearly preserved language behavior; it does not show an overall language-modeling improvement.

12.7 *From commutators to a Čech calculation*

A second KET-related study sharpens the meaning of order sensitivity. Pairwise commutators are placed on the edges of a declared finite nerve, producing a one-cochain. Hodge decomposition separates

$$c = c_{\text{exact}} + c_{\text{coexact}} + c_{\text{harm}}.$$

Only the harmonic component represents the nontrivial first cohomology class of the declared nerve.

On pretrained models, commutator energy is 86.76% exact, 10.31% coexact, and 2.92% harmonic. Task-gated repair reduces held-out closure, harmonic, and tangent non-exact RMS on all three seeds while test cross-entropy changes by -0.03%.

Design principle

Commutator energy is a local obstruction proxy. It becomes a Čech cohomology diagnostic only after a cover or nerve, orientation, cocycle test, and quotient by coboundaries have been declared.

The nerve here has vertices for attention, normalization, feedforward, and geometric incidence transport; changing the nerve changes the cohomology question. There is no claim that a transformer has one canonical Čech cover.

12.8 *Frontier-model case study: Kimi K3*

Kimi K3 provides a contemporary stress test for this architectural viewpoint. It is a 2.8-trillion-parameter, sparsely activated mixture-of-experts model, but its significance here is not its parameter count or benchmark rank. Its technical report organizes the model around three interacting forms of information flow: hybrid Kimi Delta Attention and Gated MLA across sequence positions, Attention Residuals across depth, and Stable LatentMoE across channels [Kimi Team, 2026]. The deployed system adds low-precision numerical realization and several forms of distributed parallelism. It is therefore a diagram of coupled architectural contracts rather than one undifferentiated map from tokens to tokens.

This is a retrospective LINCS reading, not a claim that Kimi K3 was designed with LINCS. The point is that a frontier architecture exposes distinct places where composition can fail:

Structural axis	Declared division of labor	Candidate obstruction
Sequence	Three recurrent KDA layers alternate with a Gated MLA layer for periodic global interaction	Long-context state is efficient but loses or misroutes information needed by global retrieval
Depth	Block Attention Residuals retrieve selected earlier representations	Compression of the depth history dilutes a representation needed downstream
Channels	A full-width shared path is combined with sparse routed experts in a lower-dimensional latent space	Routing imbalance, activation explosion, or unstable expert specialization
Numerics	Expert weights and activations use a deployment-aware low-precision realization	Quantization changes task behavior or creates a train–serve mismatch
Execution	One model is compiled into pipeline, expert, data, and context parallel operations	Partitioned execution fails to preserve the declared model within numerical tolerance

Table 12.3: A LINCS reading of Kimi K3. Each efficiency mechanism introduces a separate preservation obligation; none is certified by parameter count alone.

Quantization gives the sharpest small example. Let F_θ^{hi} denote a high-precision realization, let $F_{Q(\theta)}^{\mathrm{lo}}$ denote the deployment realization after quantizing the registered expert components, and let Φ be a task and safety observer. The relevant obstruction is not the bit width itself but the observed discrepancy

$$O_Q(x) = \Phi\left(F_\theta^{\mathrm{hi}}(x)\right) - \Phi\left(F_{Q(\theta)}^{\mathrm{lo}}(x)\right).$$

Quantization-aware post-training is then a repair intended to reduce this obstruction while retaining the deployment representation. Moonshot keeps the rollout and training quantization schemes aligned during reinforcement learning, making train–serve consistency part of the declared contract rather than an afterthought [Kimi Team, 2026].

The same discipline applies to the other axes. KDA does not make Gated MLA redundant; the hybrid declares different local and global responsibilities. Latent routing does not prove semantic preservation merely by reducing communication. Distributed execution does not become equivalent to the abstract model merely because it runs. Each claim requires its own observer, quotient, and admission evidence.

Design principle

Frontier models use compression repeatedly, but every compression is nested inside a preservation contract. Compression proposes a cheaper realization; the declared compositional obligations determine whether that realization is admissible.

12.9 *Admission contract*

Admission contract

A LINCS–KET repair is licensed only when the direct and structured routes are nonvacuous, tangent residuals pass an independent numerical check, fixed held-out probes improve blockwise, predictive semantics remain within their registered cap, and the final claim matches the task scale. A structural pass does not imply a general KET advantage or improved language modeling.

12.10 *Design lesson*

LINCS–KET is valuable precisely because the first experiment failed. The failure showed that scalarizing semantically distinct blocks can admit the wrong repair. The separately registered blockwise experiment then repaired the admission mechanism rather than erasing the unfavorable result. This is LINCS learning at the level of experimental design: non-compositionality drove not only a model update, but a better contract for deciding which updates count.

Further reading

Kan extensions and their universal properties can be approached through Mac Lane [1998] or Riehl [2017]; their use in probability is developed by van Belle [2024]. The transformer reference point is the original attention architecture [Vaswani et al., 2017]. Kimi K3 provides a current primary account of hybrid attention, latent expert routing, depth-wise residual retrieval, quantization-aware post-training, and large-scale model execution [Kimi Team, 2026]. For an accessible production-oriented overview, see Dickson [2026]; the architectural claims in this chapter follow the primary report. Diffusion models [Ho et al., 2020], diffusion language models [Li et al., 2022], and flow matching [Lipman et al., 2023] provide neighboring examples in which transport and alternative computational routes matter.

For the topology of local commutators, May [1992] supplies the simplicial background and Jiang et al. [2011] gives a statistical Hodge

decomposition on graphs. The KET paper [Mahadevan, 2026g] develops the broader categorical architecture from which the chapter's post-training LINCS audit is extracted. These readings help separate three claims that are easy to conflate: a Kan-extension specification, an implemented neural architecture, and a successful blockwise repair experiment.

13
Composable Neural Adapters

ALLORA

Low-rank adapters are attractive because they preserve a frozen backbone while adding modular capabilities. Modularity at the parameter level does not imply modularity in behavior: applying adapter i before j can differ from applying j before i. ALLORA treats that order sensitivity as a declared compositional obstruction and repairs it in the low-rank adapter space.

ALLORA abbreviates *A Lie-Algebraic Low-Rank Adaptation*. It preserves separately loadable adapters; it does not replace merging or routing baselines that may be better for a particular deployment.

13.1 *Adapters as local intervention fields*

For a frozen weight W_ℓ, a rank-r LoRA adapter contributes

$$\Delta_i^\ell = B_i^\ell A_i^\ell, \qquad W_\ell \longmapsto W_\ell + \Delta_i^\ell$$

[Hu et al., 2022]. At hidden state h, the corresponding residual map is

$$a_i^\ell(h) = h + \Delta_i^\ell h.$$

The adapter can be read as a small intervention field on representation space. For two declared-compatible adapters, the learning sketch contains

$$a_i^\ell a_j^\ell \sim a_j^\ell a_i^\ell.$$

In the linear residual realization,

$$a_i^\ell a_j^\ell - a_j^\ell a_i^\ell = \Delta_i^\ell \Delta_j^\ell - \Delta_j^\ell \Delta_i^\ell = [\Delta_i^\ell, \Delta_j^\ell].$$

Thus the matrix commutator is an exact coordinate observer of the two-route obstruction at one insertion site.

13.2 *Why the bracket controls order*

For finite strengths $\varepsilon_i, \varepsilon_j$, the Baker–Campbell–Hausdorff expansion gives

$$\exp(\varepsilon_i X_i)\exp(\varepsilon_j X_j) = \exp\left(\varepsilon_i X_i + \varepsilon_j X_j + \frac{\varepsilon_i \varepsilon_j}{2}[X_i, X_j] + O(\varepsilon^3)\right)$$

[Van-Brunt and Visser, 2016]. Reversing the order flips the leading bracket term. Hence pairwise order discrepancy is second order in adapter strength when the linear field approximation is valid.

Proposition 13.1 (Linear order bound)**.** *Let $a_i = I + \varepsilon\Delta_i$ and $a_j = I + \varepsilon\Delta_j$. Then*

$$a_i a_j - a_j a_i = \varepsilon^2[\Delta_i, \Delta_j],$$

and therefore

$$\|(a_i a_j - a_j a_i)h\| \leq \varepsilon^2 \|[\Delta_i, \Delta_j]\| \|h\|.$$

Proof. Expand both products and cancel their equal zeroth- and first-order terms. □

Across nonlinear layers, the commutator remains a local proxy rather than an exact expression for output difference. The strength-sweep experiments test where the second-order approximation remains informative.

13.3 *The ALLORA objective*

Let $\mathcal{T}_i$ denote the task loss for adapter i, and let $\mathcal{C}$ be the set of pairs declared composable. A basic realization is

$$\mathcal{L}_{\text{ALLORA}} = \sum_i \mathcal{T}_i + \lambda_c \sum_\ell \sum_{(i,j)\in\mathcal{C}} \left\| [\Delta_i^\ell, \Delta_j^\ell] \right\|_F^2 .$$

The task losses preserve independent usefulness; the bracket block reduces the selected interaction obstruction.

Definition 13.2 (ALLORA admission profile)**.** For a trained adapter family, the profile Ψ_{A} retains individual task scores, commutator norms, hidden and output order errors, and worst-order capability and safety.

The factorization gauge leaves Δ_i unchanged:

$$(B_i, A_i) \sim (B_i G, G^{-1} A_i).$$

Measuring commutators of Δ_i, rather than raw factors, makes the principal observer descend through this presentation redundancy.

13.4 Training and repair

ALLORA adapter repair

1: Train or initialize independently useful low-rank adapters.
2: Declare which adapter pairs and insertion sites should compose.
3: Measure pair and multi-adapter order sensitivity on held-out inputs.
4: Add the layerwise commutator block for declared pairs.
5: Sweep a registered range of λ_c under task-loss caps.
6: Retain independently useful adapters and evaluate both application orders.
7: Admit only Pareto-valid points; compare with merge and routing baselines.

The repair language does not require every adapter pair to commute. Some skills are intentionally order-dependent. The relation set $\mathcal{C}$ is part of the application specification.

Design principle

Regularize only interactions whose order invariance has semantic meaning. Vanishing every bracket can erase useful specialization or force adapters toward trivial behavior.

A Natural LINCS extension could equip the quotient space of adapter realizations with a Fisher or other task-semantic metric. The commutator sketch would still identify which declared interaction requires repair, while the metric would choose the least intrinsic adapter change that contracts it. This would define *Natural ALLORA*; it would not change the requirement that individual task behavior and both application orders be admitted separately. The experiments below use the declared commutator penalties directly and do not rely on this additional metric structure.

13.5 Controlled and language-model evidence

Three metrics recur below. Pair RMSE compares the outputs of $i \to j$ and $j \to i$ on the same held-out inputs. Triple RMSE aggregates disagreement across permutations of three adapters. The reported "slope" is the fitted log–log exponent of order discrepancy against a common adapter-strength multiplier; the linear proposition predicts an exponent of two near zero strength. Task Δ is always reported separately so that two equally ineffective adapter orders cannot count as a successful repair.

The controlled behavioral experiment first isolates the mechanism: at $\lambda_c = 0.03$, the bracket falls 1667×, logit-order sensitivity 1.72×10^7, and style-order sensitivity 1.89×10^6, while task metrics remain perfect.

Across a six-setting pilot corpus–backbone matrix at the same weight, commutators fall 14.5×–22.1× and output-order sensitivity falls 27.7×–77.8×. Five of six next-token losses change by less than 0.01.

Setting	Bracket	Pair RMSE	Triple RMSE	Task Δ	Slope
PTB / Transformer	21.73×	7.17×	7.31×	$+0.0005 \pm 0.0008$	1.64
WikiText-2 / KET incidence	35.46×	10.94×	11.93×	$+0.0026 \pm 0.0003$	1.71

Table 13.1: Five-seed theory-driven tests at $\lambda_c = 0.03$. Every paired seed improves; linear operator slopes are approximately 2.03, as predicted.

The network-output slopes of 1.48–1.71 are below the exact linear slope near two. This gap is useful evidence: it identifies the effect of nonlinearities and repeated insertion sites rather than pretending that the matrix model is exact end to end.

13.6 Capability and safety composition

A capability adapter for AG News classification and a safety adapter for ToxicChat moderation form a more deployment-like pair. ALLORA reduces pooled hidden order RMSE 47.7×, improves worst-order capability accuracy from 0.832 to 0.851, and safety balanced accuracy from 0.800 to 0.858. Toxic recall and benign false-positive rate move from 0.860/0.260 to 0.876/0.172.

A calibrated weighted additive merge remains stronger on average task scores: capability accuracy 0.876, safety balanced accuracy 0.873, and toxic recall/FPR 0.864/0.118. Its order sensitivity is zero by construction because it is one merged map.

Boundary

ALLORA and additive merging solve different deployment problems. The merge is a strong task-retention baseline; ALLORA is useful when capability and safety adapters must remain separately loadable and sequentially composable.

13.7 Pretrained GPT-2 probe

Experiment: registered GPT-2 Medium pilot

Across three seeds, normalized commutator falls 37.7%; pairwise logit and hidden order MSE fall 58.5% and 58.6%; triple-permutation MSE falls 57.9%. Mean task loss increases 0.0026 nats, and every geometric endpoint improves in every seed.

The pilot validates transfer of the bracket-control signature to pretrained attention adapters. Its short training schedule and low absolute SacreBLEU make it a geometric result, not evidence of high-quality semantic generation.

13.8 Structured extension: LICKET

LICKET separates an adapter into aggregation and restriction pieces,

$$\Delta_i = \Delta_i^{\mathrm{colim}} + \Delta_i^{\mathrm{lim}},$$

and penalizes cross-path interference

$$C_{ij} = [\Delta_i^{\mathrm{colim}}, \Delta_j^{\mathrm{lim}}] + [\Delta_i^{\mathrm{lim}}, \Delta_j^{\mathrm{colim}}].$$

The learned split reduces the cross ratio from 0.422 to 0.051 on WikiText-2 and from 0.387 to 0.062 on WikiText-103. In the architectural split, cross-bracket reductions exceed 400× on both corpora with perplexity within 2% of the matching baseline.

This extension illustrates the benefit of a richer sketch: instead of asking only whether whole adapters commute, it localizes which limit–colimit interaction carries the obstruction.

13.9 Admission and limitations

Admission contract

An ALLORA point is admitted only if individual adapters remain useful, declared order errors improve on held-out inputs, worst-order capability and safety clear their thresholds, and the result is competitive with merging or routing for the intended deployment. Bracket reduction alone is insufficient.

The evidence remains concentrated on small adapters, short compositions, and selected backbones. Pairwise bracket control does not guarantee benign higher-order composition, and local commutators do not capture every nonlinear network effect. The retained strength

sweeps and multi-adapter tests are therefore part of the claim, not optional illustrations.

13.10 Design lesson

ALLORA shows how a categorical declaration becomes a tractable regularizer without losing its semantics. The equation "selected adapter orders agree" comes first; the low-rank commutator observes its leading obstruction; task and safety blocks prevent trivial repair; and merge baselines reveal when sequential modularity is not worth preserving.

Further reading

The parameter-efficient lineage begins with neural adapters [Houlsby et al., 2019] and LoRA [Hu et al., 2022]. AdapterFusion [Pfeiffer et al., 2021], LoRAHub [Huang et al., 2024], and input-aware retrieval [Zhao et al., 2024] study composition and routing rather than a single isolated adapter.

Model-merging work exposes the interference problem from several angles: task arithmetic [Ilharco et al., 2023], TIES merging [Yadav et al., 2023], SVD-based LoRA merging [Tang et al., 2025], LoRA soups [Prabhakar et al., 2025], and orthogonal-subspace control [Zhang and Zhou, 2025]. Recent work on cross-task LoRA interference [Zhang et al., 2025a] is especially close in motivation. ALLORA contributes a different diagnostic: the Lie bracket measures order-sensitive interaction between adapter fields, while its admission contract retains capability and safety checks that purely algebraic merging rules may not supply.

14
Skills over Lie Algebroids

LASKO

Agentic skills do more than move a model through one state space. A skill may edit a prompt, change a schema, invoke a tool, update a validator, or alter a workflow artifact. LASKO models these controlled directions as sections of a Lie algebroid. The anchor records the visible change that a skill can execute; the bracket records order-sensitive interaction; the anchor kernel records procedural variation invisible to the final artifact.

LASKO abbreviates *Lie Algebroid Skill Optimization*. It extends the tangent-bundle interaction language of IC and ALLORA to typed, state-dependent control directions.

14.1 Why a Lie algebroid?

A tangent bundle gives every direction available at a state. An agent rarely has arbitrary access. Its possible changes are constrained by tools, permissions, schemas, and workflow state. A Lie algebroid separates the space of controlled directions from the changes they realize [Mackenzie, 2005].

Definition 14.1 (Lie algebroid)**.** A Lie algebroid over a manifold M consists of a vector bundle

$$\pi : A \longrightarrow M,$$

a Lie bracket $[\cdot,\cdot]_A$ on sections $\Gamma(A)$, and an anchor

$$\rho : A \longrightarrow TM$$

satisfying

$$[s, ft]_A = f[s,t]_A + (\rho(s)f)t$$

for sections s, t and smooth functions f.

The anchor answers "what visible edit occurs?" The bracket answers "how do two controlled edits interact?" The kernel $\ker\rho$ contains directions with no immediate visible displacement. Such directions can still alter provenance, tool state, routing assumptions, or future compositions.

Geometric object	Agent interpretation	Observable proxy	Failure
Section s	typed edit or skill policy	named repair operator	not executable in current state
Anchor $\rho(s)$	realized artifact or workflow change	diff, tool action, state transition	anchor mismatch
Bracket $[s,t]_A$	order-sensitive interaction	$s \to t$ versus $t \to s$ contrast	unstable composition
Kernel direction	latent procedural change	trace difference with similar artifact	future hidden regression

Table 14.1: The algebroid separates controlled intent, visible execution, and latent procedural state.

14.2 *The workflow learning sketch*

Let W be a partially ordered workflow and let

$$F : W \longrightarrow \mathsf{Md}$$

assign a typed Markdown artifact to each stage. Objects may include a task contract, plan graph, tool manifest, observation schema, evidence ledger, answer template, validator contract, and handoff memory.

A skill section s_i reads selected anchors and writes selected anchors. The workflow declares:

- prerequisite order between repairs;
- invariants of the tool and evidence contracts;
- restriction maps between local artifacts;
- validator paths that must agree; and
- the final artifact and trace semantics to preserve.

Definition 14.2 (LASKO residual profile)**.** For ordered sections s_i, s_j, define

$$\begin{aligned} R_{\mathrm{vis}}(i,j) &= d_{\mathrm{art}}(F_{ij}, F_{ji}), \\ R_{\mathrm{DB}}(i,j) &= d_{\mathrm{blame}}(\eta_{ij}, \eta_{ji}), \\ R_{\mathrm{ker}}(i,j) &= d_{\mathrm{trace}}(\tau_{ij}, \tau_{ji}) \quad \text{when } R_{\mathrm{vis}}(i,j) \approx 0. \end{aligned}$$

The first observer measures visible order sensitivity. The second measures whether validators assign blame differently. The third is a practical kernel proxy: the artifact may look unchanged while the route, evidence, or future repair behavior differs.

14.3 *Anchor and closure defects*

For a declared visible distribution $\mathcal{D} \subseteq TM$, LASKO keeps two geometric failures separate:

$$r_A(s,t) = [\rho(s), \rho(t)] - \rho([s,t]_A),$$

the anchor-morphism defect, and

$$r_{\text{vis}}(s,t) = (I - P_{\mathcal{D}})[\rho(s), \rho(t)],$$

the visible closure residual. The first asks whether the learned section bracket predicts executable interaction. The second asks whether the realized interaction remains inside the declared visible control span.

Boundary

A large ordered-edit contrast is not sufficient evidence for a Lie algebroid. The realization must identify typed sections, an anchor, a bracket estimator, and the Leibniz or anchor compatibilities being approximated.

14.4 *Bracket screening as repair search*

Suppose N candidate edits are available. Exhaustive validation of ordered pairs requires $N(N-1)$ expensive executions. LASKO first computes a cheap bracket proxy from read–write anchors and known workflow dependencies, then validates only the highest-scoring ordered pairs.

LASKO-prioritized repair

1. Parse the workflow into typed anchors and declared dependencies.
2. Generate admissible skill sections with read and write sets.
3. Compute cheap ordered-pair bracket or interference proxies.
4. Localize high scores to the affected workflow subdiagrams.
5. Execute only the top-ranked ordered pairs under the real validator.
6. Retain visible, blame, and trace residuals for both orders.
7. Admit a repair sequence only after held-out served validation.

The bracket is a screening device. The real validator remains authoritative. This is analogous to BRIDGE: geometry buys a smaller expensive search, while a domain evaluator chooses the final structure.

A corresponding Natural LINCS refinement would place a declared metric on the anchored skill fibers and select the least-cost section that contracts a localized bracket obstruction. This *Natural LASKO* geometry would rank tangent repair proposals, not replace the served validator or

certify that a proposed skill section integrates to a safe finite workflow edit.

14.5 Controlled benchmark

A ten-anchor Markdown benchmark contains five productive ordered chains: schema before normalization, evidence before citation, abstention before formatting, tool contract before routing, and validator before final answer. Reversing a chain loses its benefit. The normalized workflow score is

$$\text{progress} \times \text{faithfulness}.$$

Progress is the fraction of the five productive chains closed in the correct order; faithfulness checks whether the served evaluator reads the resulting skill state correctly. A bracket probe is a cheap deterministic read–write/dependency calculation. A validation probe sends a candidate artifact through the authoritative evaluator. The two columns therefore measure different resources and are never added as if they had equal cost.

Method	Bracket probes	Validation probes	Score
Greedy single edit	0	27	0.240
Random ordered pairs	0	10	0.266 mean
Exhaustive ordered pairs	0	90	1.000
LASKO-prioritized	90	10	1.000

Table 14.2: The controlled benchmark isolates the value of an interaction screen; bracket probes are cheap, validation probes expensive.

Under a served-model validation contract, exhaustive search uses 91 calls and LASKO 11; both score 1.000. Greedy and random search score 0.000 and 0.120, respectively.

14.6 Scaling the validation budget

With k independent productive chains there are $N = 2k$ edits. Exhaustive served validation grows quadratically, while LASKO validates a linear number of selected pairs.

Edits	Exhaustive calls	LASKO calls	LASKO score
10	91	11	1.000
20	381	21	1.000
40	1561	42	1.000
80	6321	84	1.000
160	25441	168	1.000

Table 14.3: Controlled scale-up with focused served validation.

This is a validation-economics result, not evidence that arbitrary agent workflows have sparse recoverable brackets. The benchmark was designed so the cheap proxy exposes the planted ordered dependencies.

14.7 From anonymous anchors to agent infrastructure

A second controlled benchmark replaces schematic anchors with task contracts, plan graphs, tool manifests, router policies, observation schemas, evidence ledgers, citation policies, validators, failure triage, and handoff memory. Six scenarios cover retrieval, extraction, code patching, research synthesis, long-horizon agency, and incident triage.

Exhaustive validation uses 133 calls and scores 1.000. LASKO uses 132 cheap bracket probes and 17 validation calls and also scores 1.000; random selection uses the same validation budget but averages 0.322. The difference is structural localization: read–write dependencies direct validation toward plausible workflow interactions.

14.8 Application probes

The broader experimental program asks whether this pattern survives outside planted chains.

10-K workflows. Staged filing pipelines provide typed evidence, mechanism, and narrative artifacts. LASKO measures whether schema, evidence, citation, and validator repairs commute.

Democritus and causal PSR. Trace-derived causal-claim workflows expose ordered edits to extraction, qualification, and validation. A predictive-state-representation (PSR) audit reaches the exhaustive served score with 72 cheap bracket probes and 19 served validations, versus 73 exhaustive validations.

SearchQA. The gated sanity check finds that a failure-derived edit can carry a stronger local interaction signal without winning hard accuracy. This is evidence for retaining soft and fiber-level audits.

ALFWorld. A micro-benchmark closes one live ordered repair edge from a failed rollout. It establishes executability of the workflow construction, not broad ALFWorld performance.

Experiment: the retained empirical hierarchy

The controlled studies establish validation savings under known sparse interaction structure. The PSR and workflow studies show that the machinery can run against served evaluators. SearchQA and ALFWorld remain scoped probes, not general agent-performance claims.

14.9 Repair admission

LASKO admission must include more than the final artifact score:

1. the section must be executable under the current tool and schema state;
2. the anchor prediction must match the realized edit;
3. the selected order must improve held-out validator behavior;
4. kernel proxies must not expose hidden trace regressions;
5. evidence and safety contracts must remain satisfied; and
6. the expensive validation savings must include all model and tool calls.

Admission contract

Cheap bracket probes may rank candidates but never replace served validation. All calls used to construct, screen, retry, or validate a repair belong in the cost certificate. Similar final Markdown is insufficient when execution traces or future compositions differ.

Design lesson

LASKO adds a critical distinction to LINCS: the space of permitted repairs is not generally the tangent bundle of the visible artifact. It is an anchored bundle of controlled, typed skills. The anchor, bracket, and kernel then separate what an agent intends, what it visibly changes, and what procedural state it leaves behind.

Further reading

The standard geometric background covers Lie groupoids and Lie algebroids [Mackenzie, 2005], together with their integrability theory [Crainic and Fernandes, 2003]. Involution algebroids and functorial semantics for Lie theory connect this geometry more directly to categorical and tangent settings [Burke and MacAdam, 2019, MacAdam, 2022].

The agentic comparison class is changing quickly. LangGraph [LangChain, 2026] provides a graph-based orchestration substrate, while SkillOpt [Yang et al., 2026] treats an external skill document as an object optimized by bounded edits and held-out validation. LASKO is complementary to such text-space optimizers: it studies typed, anchored skill sections and uses brackets to screen interaction order before expensive execution. Readers should compare full call accounting, rejected edits, transfer tests, and trace-level invariants, not only the final task score.

15
Infinitesimal Reinforcement Learning

GIRL

Temporal-difference learning turns Bellman inconsistency into a scalar error. GIRL retains the diagram that produced that error. It distinguishes the base Bellman obstruction, its tangent sensitivity, the action-common variation that should be quotiented away, the computational branch carrying the failure, and the evidence required to transport a repair across regimes.

GIRL abbreviates *Gradient Infinitesimal Reinforcement Learning*. It is an architectural layer above TD, not a replacement for the underlying policy-evaluation algorithm.

15.1 The Bellman factorization

Fix a policy π on a state space X. Let $P^\pi : \mathbb{R}^X \to \mathbb{R}^X$ denote its Markov expectation operator, let $V : \Theta \to \mathbb{R}^X$, $\theta \mapsto V_\theta$, be a parameterized value representation, and let

$$B^\pi V = R^\pi + \gamma P^\pi V$$

the Bellman operator. The GIRL sketch declares two routes from parameter space to the value-function space:

$$V_\theta \sim R^\pi + \gamma P^\pi V_\theta.$$

$$\begin{array}{ccc} \Theta & \xrightarrow{\;V\;} & \mathbb{R}^X \\ & \searrow^{B^\pi V} & \downarrow{\scriptstyle B^\pi} \\ & & \mathbb{R}^X \end{array}$$

The base obstruction at s is the Bellman residual

$$\delta_B(\theta; s) = V_\theta(s) - \big(R^\pi(s) + \gamma P^\pi V_\theta(s)\big).$$

Ordinary TD scalarizes sampled instances of this obstruction [Sutton, 1988]. GIRL tangent-lifts the same declared relation.

15.2 The tangent Bellman residual

For an admitted state direction $v \in T_s X$,

$$\delta_T(\theta; s, v) = D_s \delta_B(\theta; s)[v].$$

This asks whether the two Bellman routes respond coherently to a local change of state representation. It is not the parameter gradient of the scalar TD loss.

Definition 15.1 (GIRL architectural interface)**.** A GIRL realization exposes:

1. the base Bellman residual δ_B;
2. the tangent residual δ_T on declared probes;
3. a cover of the transition, reward, bootstrap, and representation branches;
4. a decision-relevant quotient; and
5. an admission controller for cross-regime use.

For isotropic random probes with $\mathbb{E}[vv^\top] = I$, writing $J_s = D_s \delta_B$ gives

$$\mathbb{E}_v \|J_s v\|^2 = \operatorname{tr}(J_s^\top J_s) = \|J_s\|_F^2.$$

JVPs therefore estimate the squared tangent residual without materializing a full Jacobian.

15.3 IL–GTD–MP

In a linear off-policy realization, the infinitesimal-learning variant of gradient temporal difference Mirror–Prox (IL–GTD–MP) augments the GTD–Mirror–Prox objective with a quadratic tangent Bellman penalty [Sutton et al., 2008, Maei, 2011, Nemirovski, 2004].

Proposition 15.2 (Quadratic tangent penalty)**.** *For a linear value model and fixed probe covariance, the expected squared tangent Bellman residual is a positive-semidefinite quadratic function of the value parameters. Adding it to the regularized GTD saddle objective preserves monotonicity of the associated variational inequality.*

Proof. The Bellman residual derivative is affine in the linear parameters. Its squared norm therefore has a positive-semidefinite Gram Hessian. Adding that term to a monotone regularized GTD operator preserves monotonicity. □

Standard stochastic Mirror–Prox rates then apply under the usual boundedness, variance, and step-size hypotheses. The guarantee concerns optimization of the declared linear objective. It does not establish global robustness or policy improvement.

Boundary

A small tangent residual controls first-order variation only on the support and in the metric used to define it. Finite-shift and policy claims additionally need coverage, derivative regularity, Bellman inverse conditioning, and a positive action gap.

15.4 *Why the advantage quotient is necessary*

Absolute action values can vary smoothly while their ranking is wrong. For policy learning, GIRL quotients the action-common direction:

$$A_\theta(s,a) = Q_\theta(s,a) - \sum_b \pi(b \mid s) Q_\theta(s,b),$$
$$C_\theta(s;a,a') = Q_\theta(s,a) - Q_\theta(s,a').$$

Adding the same $c(s)$ to every action leaves C_θ unchanged. The tangent contrast regularizer

$$\mathcal{R}_A(\theta) = \mathbb{E}_{s,a,a'}\left[\|D_s C_\theta(s;a,a')\|^2\right]$$

therefore descends to the decision-relevant quotient.

Design principle

Probe the information used by the decision rule. Smooth absolute values are not enough when only action contrasts determine the policy.

15.5 *Natural GIRL*

The advantage quotient identifies which policy distinctions matter, but it does not yet choose an intrinsic repair direction. Let Π be a smooth policy object, let $\mathcal{O}_{\mathrm{G}}(\pi)$ denote the typed GIRL obstruction, and let F_π be a policy Fisher metric defined using a declared state-visitation law. The Natural LINCS construction of Chapter 4 specializes to

$$\xi_\pi^\star \in \operatorname*{arg\,min}_{\xi \in T_\pi \Pi} \frac{1}{2} F_\pi(\xi,\xi) \quad \text{subject to} \quad D\mathcal{O}_{\mathrm{G},\pi}[\xi] = -\mathcal{O}_{\mathrm{G}}(\pi). \tag{15.1}$$

This is *Natural GIRL*: the Bellman, intervention, or advantage sketch defines what requires repair, while information geometry selects the

least intrinsic policy change that cancels the linearized obstruction. For a scalar policy objective, the construction reduces in coordinates to the familiar natural policy-gradient direction. Natural actor–critic supplies one family of estimators for that direction; it does not replace the GIRL declaration or admission controller.

The visitation law in F_π is semantic data. On-policy, discounted, stationary, replay-weighted, and target-policy distributions generally define different geometries. In the off-policy setting of this chapter, the metric declaration must therefore be audited together with coverage and effective sample size. A Fisher matrix may also be singular when distinct parameter directions induce the same policy. Such directions should be quotiented as policy gauges before inversion, or handled by a registered damping or pseudoinverse rule.

Design principle

GIRL declares the decision-relevant obstruction; Natural GIRL chooses an intrinsic tangent proposal; the safety and recovery controller decides whether that proposal may alter the policy.

15.6 *Localization over Bellman covers*

The Bellman computation can be covered by local subdiagrams for representation, transition, reward, successor value, and action contrast. Shared bootstrap branches form overlaps.

Proposition 15.3 (Tangent stability of Bellman localization)**.** *For every cover inclusion $i_a : J_a \to J$,*

$$T(D \circ i_a) = (TD) \circ i_a.$$

If the cover is complete and tangent lift preserves its overlap descent data, compatible local tangent factorizations glue to a global tangent factorization.

Proof. The restriction identity is functoriality of T. Under the stated descent hypotheses, the overlap equalities that glue the base family also glue its tangent lift. □

Localization of obstruction data does not imply localization of parameter repair. Cotangent transport and parameter sharing can couple distant modules.

15.7 *Six layers of control*

GIRL separates six obligations:

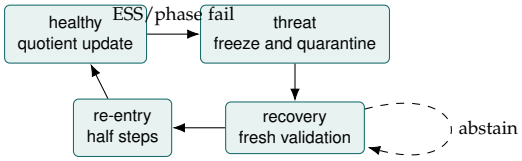
1. **Advantage quotient**: remove action-common variation.

2. **Phase coherence**: check that validation and deployment directions align.
3. **Confidence**: require a positive one-sided utility bound.
4. **Successor sampling**: estimate the Bellman branch without privileged population operators.
5. **ESS adaptation**: select an off-policy estimator from replay support.
6. **Temporal control**: quarantine stale experience and govern re-entry.

For normalized validation and deployment directions,

$$\kappa = \langle d_{\text{val}}, d_{\text{dep}} \rangle$$

is the phase-coherence gate. At high effective sample size, ordinary importance sampling can be efficient; under severe mismatch, clipped self-normalization can be safer.



15.8 The retained experimental sequence

The experiments were designed as a sequence in which failures determine the next structural layer. In the table, regret measures the value gap induced by the selected action, and ranking accuracy measures whether action contrasts have the correct order. A harmful admission is an update whose independently computed exact utility is negative. Exact utility is used only to score the experiment; the controller sees probe, confidence, phase, ESS, and temporal-state information. Post-shift AUC integrates return over the evaluation epochs following the hidden regime change.

Stage	Retained result	Failure isolated	Layer added
Value-level IL–GTD–MP	tangent RMSE -24.1%; OOD Bellman RMSE -81.8%	does not imply policy improvement	action-conditioned test
Full-state policy bridge	regret gain only 0.073; smooth wrong control passes	absolute values hide rankings	advantage quotient
Advantage quotient	regret $3.465 \to 2.081$; ranking $+12.6$ points	orthogonal phase failures	coherence gate
Sampled transport	zero harmful admissions; 95.0% high-budget retention	uncertainty causes abstention	confidence block
Off-policy transport	zero harm but registered $74.84\% < 75\%$ threshold	estimator depends on mismatch	ESS meta-gate
Adversarial online shift	unconditional harm 22.67%; router zero harm	freezing delays recovery	temporal recovery

Table 15.1: Scientific failures, technical misses, and negative controls remain visible; each motivates a different architectural change.

15.9 *Safe recovery*

In a 12-state online actor–critic, the benign control contains no safety pathology: replay ESS remains near 0.96, no method makes a harmful update, and unconditional quotient GIRL achieves the best final return. Always-on gating is unnecessarily conservative. This negative result motivates routing: use the quotient alone while diagnostics are healthy.

Experiment: GIRL safe recovery under hidden regime change

In the adversarial study, reward preference and action-transition kernels change at epoch 10 while stale replay remains. The shift time is hidden from the methods. Unconditional GIRL makes harmful updates in roughly 23% of post-shift epochs. The recovery controller detects the shift in every seed, reduces harm to zero, and re-enters in 29/30 seeds after 4.14 epochs on average.

Method	Post-shift AUC	Final return	Harm
Unconditional	-1.093	-0.299	23.0%
Freeze router	-1.255	-0.759	0%
Recovery router	$\mathbf{-0.911}$	$\mathbf{-0.019}$	**0**%

Table 15.2: Locked safe-recovery run over seeds 270–299.

One seed remains abstinent through the finite horizon. The correct interpretation is unresolved uncertainty, not an unrecoverable environment.

15.10 *Global-to-local reconstruction*

In a two-block dueling deep Q-network (DQN), the zero-inclusive global phase preserves return but reduces tangent obstruction by only

3.92%, below the registered 5% gate. Critic-only local heads reconstruct only about 54% of the global obstruction variance. Adding the shared Bellman overlap $(y, D_s y\, v)$ raises reconstruction R^2 to 0.989/0.994 and reduces normalized gluing error to 0.0057. Shuffling that overlap destroys the result.

Local/global repair-gradient cosines remain only 0.464/0.584. The study therefore supports localization of obstruction data, not yet compositional local repair.

Admission contract

A GIRL update is admitted only after quotient validity, probe semantics, coverage, phase coherence, uncertainty, replay support, and temporal state are checked. Exact returns may score harm in the experiment but must not leak into the controller decision.

Design lesson

GIRL shows why a tangent signal is not one more regularizer. The full-state signal fails scientifically; the advantage quotient exposes the decision-relevant object; transport and ESS gates separate geometry from statistics; and temporal quarantine turns abstention into safe recovery. The learning occurs in the controller and declaration as much as in the value parameters.

Further reading

The reinforcement-learning foundation is Sutton and Barto [2018]. Temporal difference learning begins with Sutton [1988]; gradient-TD methods and their off-policy motivation are developed in Sutton et al. [2008], Maei [2011]. Stochastic Mirror–Prox [Nemirovski, 2004, Juditsky et al., 2011] supplies the optimization background for the linear saddle realization.

For decision-relevant credit, compare hindsight credit assignment [Harutyunyan et al., 2019], return decomposition [Arjona-Medina et al., 2019], and policy-level trust regions [Schulman et al., 2015]. Natural policy gradient [Kakade, 2001] and natural actor–critic [Peters and Schaal, 2008] provide the information-geometric background for Natural GIRL. Safe policy improvement under limited coverage is a neighboring concern; decision-point restriction provides a recent example [Sharma et al., 2025]. GIRL differs by tangent-lifting the Bellman factorization and quotienting action-common variation before admission, but the safe and offline RL literature supplies essential baselines for coverage, distribution shift, pessimism, and high-confidence improvement.

Part III

Relational Learning and Reasoning

16
Learning from Structured Preferences

LINCS–RLHF

RLHF usually turns pairwise preferences into a scalar reward and then optimizes that reward. LINCS–RLHF treats the scalar reward as a *factorization hypothesis*. If comparison log odds circulate around a cycle, no global scalar potential represents the declared preference relation. More reward optimization cannot repair a representation that does not exist.

The structural question comes before optimizer choice: should the preference relation be represented by a scalar potential at all?

16.1 The scalar-reward declaration

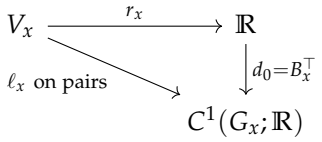
Fix a prompt or context x. Let $G_x = (V_x, E_x)$ be a connected comparison graph whose vertices are responses. Choose an orientation and let $B_x \in \mathbb{R}^{|V_x| \times |E_x|}$ be its signed incidence matrix. If $H_x(i,j)$ is the probability that response i is preferred to response j, define the edge log odds

$$\ell_x(i,j) = \operatorname{logit} H_x(i,j).$$

A Bradley–Terry reward $r_x : V_x \to \mathbb{R}$ declares

$$\ell_x = B_x^\top r_x.$$

This is the representational assumption shared by explicit reward modeling and preference objectives built from reward differences [Bradley and Terry, 1952, Christiano et al., 2017, Rafailov et al., 2023].



Let the columns of Z_x form a basis of the cycle space $\ker B_x$.

Definition 16.1 (Scalarization obstruction)**.** The obstruction to scalar reward representation is

$$\Omega_x = Z_x^\top \ell_x,$$

or, independently of a chosen basis,

$$[\ell_x] \in C^1(G_x; \mathbb{R}) / \operatorname{im} d_0.$$

Theorem 16.2 (Cycle criterion for reward scalarization)**.** *For a connected comparison graph, the following are equivalent:*

1. $\ell_x = B_x^\top r_x$ *for some scalar reward* r_x*;*
2. $z^\top \ell_x = 0$ *for every cycle flow* $z \in \ker B_x$*; and*
3. $Z_x^\top \ell_x = 0$ *for one, hence every, cycle basis.*

When these conditions hold, r_x *is unique up to an additive constant.*

Proof. The fundamental theorem of linear algebra gives

$$\operatorname{im} B_x^\top = (\ker B_x)^\perp.$$

Connectedness leaves only the constant vector in $\ker B_x^\top$, which is the reward gauge. □

For a triangle i, j, k, the witness is simply

$$\Omega_x(i, j, k) = \ell_x(i, j) + \ell_x(j, k) + \ell_x(k, i).$$

This is the cycle component in the graph-Hodge decomposition of pairwise rankings [Jiang et al., 2011]. LINCS uses it as the obstruction to a declared downstream representation, rather than merely as another fit statistic.

Boundary

Vanishing circulation certifies scalarizability only on the graph that was observed. A tree has no cycle witness: every observed field admits a potential, yet unobserved comparisons can complete it to either a scalar or a cyclic field. "No witness" is unidentifiable, not accepted.

16.2 *Projection is not repair*

With positive edge weights W_x, a scalar model can always return the weighted projection

$$r_x^* \in \arg\min_{\mathbf{1}^\top r = 0} \left\| W_x^{1/2} (\ell_x - B_x^\top r) \right\|^2.$$

The residual $h_x = \ell_x - B_x^\top r_x^*$ is the weighted cycle component. Better optimization can reduce estimation error inside the gradient subspace; it cannot eliminate a population component outside that subspace. Reward-model fit and reward representability are therefore different questions.

16.3 Tangent transport and reward gauge

Suppose $\ell_{x,\theta}$ is differentiable and a declared perturbation $\dot{\theta}$ induces $\dot{\ell}_x = J_{\ell_x}(\theta)\dot{\theta}$. The tangent obstruction is

$$T\Omega_x(\dot{\theta}) = Z_x^\top J_{\ell_x}(\theta)\dot{\theta}.$$

It distinguishes directions that preserve, create, or repair circulation, even when their base fields agree.

Proposition 16.3 (Naturality and gauge)**.** *The base and tangent obstructions are invariant under admissible response relabeling. They are also unchanged by $r_x \mapsto r_x + c_x\mathbf{1}$.*

Proof. Signed edge permutation transports B_x, Z_x, and ℓ_x equivariantly. The resulting permutation matrices cancel in $Z_x^\top \ell_x$ and its derivative. Gauge invariance follows from $B_x^\top \mathbf{1} = 0$. □

Prompt-common reward offsets are thus quotiented as representational gauge. They cannot change a pairwise decision and should receive no structural credit.

16.4 Localize before pooling

Let $s = (x, a, g)$ index a prompt, annotator population, or other declared stratum. The localized obstruction is the family

$$\Omega_{\mathcal{U}} = \{Z_s^\top \ell_s : s \in \mathcal{U}\}.$$

Aggregation can erase precisely the variation that matters.

Proposition 16.4 (Opposite-cycle cancellation)**.** *Let one population have reciprocal comparison probabilities with log odds ℓ and nonzero circulation. A second population reverses every comparison and therefore has log odds $-\ell$. Their equal probability mixture has zero pooled log odds on every edge, although neither population is scalarizable.*

Proof. If $H = \sigma(\ell)$, reversal gives $\sigma(-\ell) = 1 - H$. The equal mixture is $1/2$, whose logit is zero, while the two local circulations remain opposite and nonzero. □

Design principle

The cover declares where a representation must be valid. Pooling is permitted only after local obstructions and their overlap semantics agree.

16.5 Statistical admission

For estimated log odds $\widehat{\ell}_s$ with covariance $\widehat{\Sigma}_s$, set

$$\widehat{\Omega}_s = Z_s^\top \widehat{\ell}_s, \qquad \widehat{V}_s = Z_s^\top \widehat{\Sigma}_s Z_s.$$

Under the scalarizable null and the usual fixed-graph asymptotics,

$$Q_s = \widehat{\Omega}_s^\top \widehat{V}_s^\dagger \widehat{\Omega}_s \implies \chi^2_d,$$

where $d = \operatorname{rank} V_s$. This yields a three-way decision:

$$\mathcal{D}(G_s, \widehat{\ell}_s) = \begin{cases} \mathsf{U}, & d = 0, \\ \mathsf{R}, & d > 0 \text{ and } p_Q \le \alpha, \\ \mathsf{A}_G, & d > 0 \text{ and } p_Q > \alpha. \end{cases}$$

Here U requests more comparisons or a weaker target, R rejects scalarization, and A_G conditionally admits it on observed support. Nonrejection is not proof of a scalar population reward.

16.6 Relational fallback

When scalarization is rejected, retain the centered reciprocal game

$$A = H - \tfrac{1}{2}\mathbf{1}\mathbf{1}^\top, \qquad A^\top = -A,$$

and solve

$$\pi^* \in \arg\min_{\pi \in \Delta(V)} \max_{q \in \Delta(V)} q^\top A \pi.$$

Proposition 16.5 (Relational minimax guarantee)**.** *For every reciprocal population matrix H, the game value is zero and a population minimax policy has zero pairwise exploitability. Uniform play is minimax when $A\mathbf{1} = 0$, but need not be minimax in asymmetric games.*

Proof. Skew symmetry gives $\pi^\top A \pi = 0$. The finite minimax theorem and exchange of the identical strategy simplices negate the value, so it equals zero. □

The guarantee does not eliminate finite-sample error: an empirical minimax policy can still be exploitable under the population game.

16.7 What the experiments establish

Experiment: LINCS-RLHF controlled preference suite

E1–E4 are controlled preference simulations in which independent random seeds are the replication unit and repeated pairwise labels are sampled from a known reciprocal population matrix. E1 varies labels per edge and cyclic strength to separate null calibration from power. E2 assigns opposite cycles to two annotator strata and compares pooled with localized decisions. E3 uses complete five- and seven-response games and evaluates fitted policies against the population matrix. E4 varies observed graph support among trees, one-cycle graphs, sparse connected graphs, and complete graphs. Oracle population quantities score detection and exploitability but are unavailable to the fitted gate and policies.

Study	Retained result	Failure or boundary	Architectural consequence
E1: calibration and power	null calibration within 0.015 for $n \geq 40$; power at least 0.930 in the registered high-information regime	near-zero power in deliberately weak cells	admit only at declared support and resolution
E2: opposite populations	equal-mixture pooling falsely admits 99.6%; localization changes mean exploitability $0.089 \to 0.047$	four low-information cells worsen by at most 0.0042	preserve annotator strata before aggregation
E3: larger games	mean exploitability: scalar 0.084, uniform fallback 0.094, relational fallback 0.026	55 of 144 every-cell criteria fail, mostly balanced games	relational fallback is preferred, not universally dominant
E4: incomplete graphs	all trees labeled unidentifiable; false rejection 0.008 at $\alpha = .01$ on cyclic-support graphs	power depends on signal alignment, not cycle rank alone	collect repeated comparisons on informative cycles

Table 16.1: The registered sequence separates calibration, localization, fallback quality, and graph identifiability. Negative cells remain visible.

A public-data audit sharpens the collection requirement. OpenAI summarization-feedback batch 3 contained prompt-level cycles after support thresholding, but no worker-by-prompt stratum retained a cycle once each edge required even two labels. It can support a prompt-level feasibility analysis, not the annotator-local mechanism tested in E2. An audit-ready preference dataset must preserve stable response identities, repeated connected cycles, and annotator metadata.

Admission contract

Scalar policy optimization is admitted only when the declared comparison graph contains estimable cycle witnesses, the stratum-level test does not reject at its frozen threshold, support covers the intended deployment graph, and multiplicity is handled at the level of the scientific claim. Otherwise the controller collects data, weakens the target, abstains, or retains the relational game.

Design lesson

LINCS–RLHF separates learning preferences from forcing them into rewards. Circulation diagnoses whether scalarization is structurally available; tangent transport identifies how that availability changes; localization prevents heterogeneous populations from canceling; and admission controls which representation reaches policy optimization. When preferences do not scalarize, the repair is to retain relational structure, not to optimize its scalar projection harder.

Further reading

The standard scalar pipeline combines preference-based reinforcement learning [Christiano et al., 2017] with Bradley–Terry comparison models [Bradley and Terry, 1952]; direct preference optimization retains the same reward-difference form inside a policy objective [Rafailov et al., 2023]. Reward overoptimization [Gao et al., 2023] concerns what happens when a learned scalar proxy is optimized too strongly. LINCS–RLHF asks the logically prior question of whether the comparison field is representable by any scalar reward on the declared support.

Graph-Hodge ranking [Jiang et al., 2011] supplies the gradient/cycle decomposition. Relational alternatives include minimax RLHF [Swamy et al., 2024] and general preference models [Zhang et al., 2025b]. Recent hybrid work explicitly separates transitive and cyclic components and optimizes the resulting game dynamically [Huang et al., 2026]. Together these sources suggest a growing literature in which scalar reward, cyclic structure, population heterogeneity, and game solutions are treated as distinct modeling choices rather than forced into one ranking.

17
Relational Manifold Recovery

RADAR

Relational data do not begin as point clouds. Tables, typed edges, foreign keys, and join witnesses form a diagram before they form any homogeneous proximity graph. RADAR retains that diagram, designates its most informative view as an apex, and uses LINCS to repair compatibility with simpler faces without trading away their private geometry.

RADAR abbreviates *Relational Apex Descent for Auditable Realizations*. The apex is the view that retains declared information its faces forget; it is not merely the best-scoring encoder.

17.1 A database is a diagram

Let $\mathcal{S}$ be a category presenting a database schema. A database instance is a functor

$$I : \mathcal{S} \longrightarrow \mathbf{Set}$$

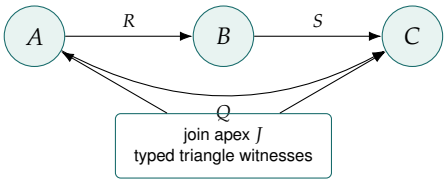
that assigns rows to entity types and functions to declared foreign keys [Spivak, 2012]. In the controlled construction,

$$R \subseteq A \times B, \qquad S \subseteq B \times C, \qquad Q \subseteq A \times C.$$

The higher-order evidence is the set of join witnesses

$$J = \{(a, b, c) : (a, b) \in R,\ (b, c) \in S,\ (a, c) \in Q\}.$$

A witness records which three typed facts cohere. It is not simply another pairwise similarity.



Proposition 17.1 (Projection non-identifiability)**.** *Let P map database instances to weighted graphs on A. If $P(I_0) = P(I_1)$ while I_0 and I_1 have*

different join-witness structure, every deterministic embedding that factors through P returns the same output on both. Under equal priors, no evaluator of that output can identify the generating join structure with accuracy above $1/2$.

Proof. For an embedding E, equality of $P(I_0)$ and $P(I_1)$ implies equality of $E \circ P$. Conditioning a randomized method on its seed gives the same argument. □

This is not a criticism of Uniform Manifold Approximation and Projection (UMAP) [McInnes et al., 2018]; it applies to every method given the same lossy projection. A triangle-feature UMAP baseline is therefore necessary: it separates access to higher-order information from the value of relational descent.

17.2 *Foundries and the apex*

Let $\mathcal{U} = \{U_i\}$ be typed database views. Foundry i owns:

1. a carrier V_i of database keys and provenance;
2. a local section $z_i : V_i \to \mathbb{R}^2$;
3. a private fuzzy graph $G_i = (E_i, \mu_i)$; and
4. a restriction interface declaring what may be compared with other views.

For every private edge $e = (p, q)$, the initial distance determines a band

$$\ell_{ie} \leq \|z_i(p) - z_i(q)\| \leq u_{ie}.$$

A signed anchor triangle prevents rank collapse:

$$\sigma_i \det[z_i(b_i) - z_i(a_i), z_i(c_i) - z_i(a_i)] \geq \rho_i.$$

Let $y_v \in \mathbb{R}^2$ be a global coordinate and $H_i = (R_i, t_i) \in SE(2)$ carry foundry i into the global frame. Incidence compatibility is

$$g_{iv} = R_i z_i(v) + t_i - y_v = 0.$$

The join-aware geometric-transformer view initializes y; the R and Q faces are aligned to it. Symmetric averaging would incorrectly declare the information-forgetting faces equivalent to their apex.

Definition 17.2 (Admissible relational realization)**.** An admissible realization satisfies every declared incidence, private fuzzy band, non-collapse anchor, chart-centering constraint, and global Euclidean gauge. When learned foundries disagree, the terminal incidence obstruction is reported rather than hidden inside a weighted embedding objective.

17.3 Infinitesimal relational descent

Writing $K = \left[\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right]$, the tangent incidence equation is

$$\delta g_{iv} = R_i K z_i(v)\, \delta\alpha_i + R_i \delta z_i(v) + \delta t_i - \delta y_v.$$

Stacking these rows gives a sparse Jacobian Dg. The computational repair solves

$$Dg\, \delta = -g$$

in a damped metric. This selects a tangent direction; it does not define the semantic meaning of compatibility.

Private boundaries are activated before they are crossed. For $s_{ie} = \|z_i(p) - z_i(q)\|^2$,

$$\dot{s}_{ie} = 2(z_i(p) - z_i(q))^\top (\delta z_i(p) - \delta z_i(q)).$$

If the proposed direction crosses an edge band or anchor-area floor, its normal covector enters the active solve. Backtracking then evaluates the exact nonlinear declarations.

RADAR relational descent

1: Construct typed face and join-witness foundries
2: Compute centered local sections and deterministic orientations
3: Build private fuzzy bands and signed-area anchors
4: Initialize the global section from the apex; align its faces
5: **while** an improving admissible step exists **do**
6: Linearize incidence and gauge equalities
7: Solve the damped sparse tangent system
8: Activate threatened edge and area faces; re-solve
9: Clip units separately and backtrack against the exact audit
10: Retract centering and the global $SE(2)$ gauge exactly
11: **end while**
12: Return the realization, rejected steps, and terminal obstruction

---

17.4 What is exact

Define the transition between foundry frames by

$$G_{ji} = H_j^{-1} H_i.$$

Proposition 17.3 (Exact induced cocycles)**.** *For every triple overlap,*

$$G_{ki} = G_{kj} G_{ji}.$$

Triple consistency therefore requires no separate cycle loss.

Proof. $(H_k^{-1}H_j)(H_j^{-1}H_i) = H_k^{-1}H_i$. □

This exact transition cocycle does not imply small incidence obstruction. Frame consistency and agreement of the local sections are distinct obligations.

Proposition 17.4 (Feasibility and apex stability)**.** *If each candidate is accepted only after exact nonlinear auditing, every accepted iterate remains feasible to the fixed tolerance and strictly lowers incidence RMS. Moreover, if an apex linear probe has margin γ and the aligned repair displaces every point by less than $\gamma/\|w\|$, its predicted labels are unchanged.*

Proof. Feasibility follows inductively from the acceptance predicate. The margin claim is Cauchy–Schwarz: no signed score can move by γ. □

Design principle

RADAR does not ask a scalar loss to trade incidence against private geometry. The diagram defines a vector obstruction and a feasible set; the solver chooses a direction, and the exact declarations decide whether it may enter.

17.5 Controlled relational recovery

Closed and twisted synthetic databases have similar pairwise relation graphs but different join closure. Ten paired seeds compare flattened PCA, several UMAP and Hodge projections, edge- and triangle-aware geometric transformers, and RADAR. Labels never construct a representation or select a repair. Every method receives exactly the information named in its row: pairwise baselines receive flattened relations, the triangle baseline receives engineered join features, and apex methods receive typed witnesses. A common downstream linear probe scores the resulting coordinates. Accuracy therefore measures information retained under a fixed two-dimensional bottleneck, while incidence, private-edge, anchor, and cocycle audits determine whether the RADAR realization itself is admissible.

Representation	Accuracy	ROC AUC	Interpretation
PCA, flattened relations	0.485 ± 0.029	0.512 ± 0.035	homogeneous feature projection
UMAP, R-Jaccard	0.506 ± 0.029	0.490 ± 0.040	pairwise face only
UMAP, triangle features, 2D	0.545 ± 0.060	0.568 ± 0.101	higher-order features under the same bottleneck
Pooled Hodge L_1	0.519 ± 0.011	0.499 ± 0.051	fixed typed-complex comparator
GT apex, PCA to 2D	$\mathbf{0.906 \pm 0.014}$	0.905 ± 0.015	join-aware apex before repair
RADAR	$\mathbf{0.906 \pm 0.014}$	$\mathbf{0.905 \pm 0.012}$	compatible audited realization

Table 17.1: Primary endpoint at mode-1 closure probability 0.15. RADAR preserves the apex signal; it does not improve it using hidden labels.

Experiment: RADAR two-dimensional recovery

Against R-only UMAP, the paired accuracy gain is 0.400, with bootstrap interval $[0.380, 0.421]$. Across the complete 50-run noise sweep, incidence falls in every run; maximum private-edge violation is 9.34×10^{-9}, minimum anchor retention is 99.22%, and induced cocycle RMS is at most 1.66×10^{-16}.

Boundary

The capacity diagnostic is deliberately retained. Triangle-feature UMAP reaches 0.906 accuracy in 8D and 16D. The result is therefore an audited two-dimensional recovery under a severe bottleneck, not an information-theoretic separation from every engineered graph representation.

17.6 *MovieLens as a conjunctive gate*

Experiment: RADAR MovieLens conjunctive gate

The chronological MovieLens 100K study uses only training events to construct geometry and joins [Harper and Konstan, 2015]. Apex RADAR exceeds matched rating-weighted UMAP by 0.0272 normalized discounted cumulative gain at rank ten (NDCG@10), reduces incidence in all five seeds, retains at least 98.4% of every anchor, and closes cocycles to numerical precision. Those structural and ranking gates pass.

Its rating root-mean-square error (RMSE) is 1.154, however, which is 0.154 worse than the 2D matrix-factorization comparator and fails the preregistered 0.05 safeguard. The correct outcome is diagnostic, not a positive recommendation claim.

Admission contract

A RADAR realization is promoted only when incidence strictly decreases, private bands and non-collapse anchors remain feasible, induced cocycles close, the apex signal survives, and every task-specific safeguard passes. Passing ranking while failing calibration blocks the aggregate claim.

Design lesson

RADAR separates three sources of value. The apex encoder supplies higher-order information; LINCS supplies compatibility repair; exact audits determine admission. The method cannot reconstruct witnesses discarded before the apex was built, and compatibility alone does not guarantee downstream calibration. Its contribution is an auditable relational realization whose successes and terminal failures remain typed.

Further reading

The categorical database foundation is functorial data migration [Spivak, 2012]. For manifold learning, UMAP [McInnes et al., 2018], PaCMAP [Wang et al., 2021], and densMAP [Narayan et al., 2021] illustrate different choices about neighborhood, global, and density preservation. These methods begin after data have been presented as points or a proximity structure; RADAR asks what typed relational information is lost in that presentation.

Relational graph convolution [Schlichtkrull et al., 2018], simplicial neural networks [Ebli et al., 2020], cellular networks [Bodnar et al., 2021], and generalized simplicial attention [Battiloro et al., 2024] show how typed edges and higher cells can enter a learned representation. Statistical Hodge theory [Jiang et al., 2011] provides a complementary spectral decomposition. RADAR should be compared with these methods under matched information access and output dimension; its distinctive claim is exact, apex-directed compatibility auditing rather than universal superiority of relational geometry.

18
Sheaf-Based Distributed Learning

SID

Foundries maintain reusable local knowledge across contexts. SID instantiates them as sheaf-valued predictive models and makes the descent diagram itself the learning object. It separates local compatibility, global effectivity, tangent updates that preserve descent, and transverse repairs that return an obstructed family toward the descent locus.

SID abbreviates *Sheaf Infinitesimal Descent*. It replaces a representation-specific gluing penalty with a typed descent-and-admission interface.

18.1 Compatibility and effectivity

Let $\{U_i \to U\}$ be a finite cover and let a presheaf X assign a predictive-state object $X(U_i)$ to each context. Restrictions expose shared questions:

$$\rho^i_{ij} : X(U_i) \longrightarrow X(U_{ij}).$$

Local sections $x_i \in X(U_i)$ are compatible when

$$\rho^i_{ij}(x_i) = \rho^j_{ij}(x_j)$$

on every declared overlap. Predictive-state realizations are natural here because they describe a system through observable tests rather than a single assumed latent state [Littman et al., 2001, Singh et al., 2004b].

Define

$$M_{\mathcal{U}} = \prod_i X(U_i), \qquad O_{\mathcal{U}} = \prod_{i,j} X(U_{ij}),$$

and let $d_0, d_1 : M_{\mathcal{U}} \rightrightarrows O_{\mathcal{U}}$ apply the two restrictions. The compatible-family object is

$$Z_{\mathcal{U}} = \mathrm{Eq}(d_0, d_1).$$

A separate map

$$\eta_{\mathcal{U}} : X(U) \longrightarrow Z_{\mathcal{U}}$$

restricts a declared global model to its local family.

$$X(U) \xrightarrow{\eta_{\mathcal{U}}} Z_{\mathcal{U}} \hookrightarrow M_{\mathcal{U}} \underset{d_1}{\overset{d_0}{\rightrightarrows}} O_{\mathcal{U}}$$

Compatibility asks whether a family lies in $Z_{\mathcal{U}}$. Effectivity asks whether it lies in the image of $\eta_{\mathcal{U}}$. Local models can agree on every measured overlap while failing rank, normalization, causal, or dynamical constraints of the global model class.

Definition 18.1 (SID factorization problem)**.** The base SID obstruction is the obstruction to inhabiting both the compatibility and effectivity factorizations of the declared descent sketch. The tangent obstruction is the same factorization problem after applying the tangent functor to its objects, restrictions, and cones.

Design principle

Compatibility and effectivity are two typed obstructions, not two terms in one loss. They can demand different repairs: transport between local models, expansion of the global class, a new context, or refusal to promote.

18.2 *Why a gluing loss is not the declaration*

Earlier Prometheus and Odyssey realizations used a weighted gluing tension

$$\mathcal{L}_{\text{glue}} = \sum_{i,j} w_{ij} \left\| \rho^i_{ij}(x_i) - \rho^j_{ij}(x_j) \right\|^2$$

[Mahadevan, 2026k,j]. It is a useful concrete architecture, but it presupposes subtraction, a norm, and a weight relative to prediction. It also vanishes on every compatible family, including families outside the admitted global class.

SID retains the cover, provenance, restrictions, and promotion semantics but makes the descent factorization primitive. A metric may choose among corrections; it does not define what gluing means.

Proposition 18.2 (Restriction to a subcover)**.** *A global compatibility witness restricts naturally to every declared subcover, before and after tangent lift. Conversely, vanishing on selected subcovers implies global compatibility only if they jointly contain every declared simplex and their corrections agree on shared local objects.*

Independently repairing each pair can assign incompatible updates to the same foundry. Localization must itself be a compatible family.

18.3 *The linear predictive-state model*

Let $\theta_i \in \mathbb{R}^{p_i}$ parameterize foundry i, and let R^i_{ij} select or transport predictions to shared tests. After choosing an orientation of the overlap

graph, stack the restriction differences:

$$A\theta = \left(R^i_{ij}\theta_i - R^j_{ij}\theta_j\right)_{(i,j)}.$$

The compatible locus is $\ker A$. If u is a proposed predictive update, the unit SID correction solves

$$Aa = -A(\theta + u).$$

When A has full row rank, the minimum-norm realization is

$$a = -A^\top (AA^\top)^{-1} A(\theta + u).$$

Theorem 18.3 (Compatibility and prediction decomposition)**.** *Let*

$$P_A = I - A^\top (AA^\top)^{-1} A.$$

Starting from a compatible θ, the repaired update is

$$\theta^+ = \theta + P_A u.$$

Thus SID retains precisely the component of the predictive update tangent to the compatibility subspace. From an arbitrary state, the unit repair projects $\theta + u$ onto $\ker A$.

Proof. Substitute the displayed correction and use $A\theta = 0$. The matrix P_A is the orthogonal projector onto $\ker A$. □

The theorem also marks a boundary: projection may remove a locally useful component of u. Structural and predictive metrics must be reported separately.

18.4 Tangent descent

Assume the predictive category carries tangent structure [Cockett and Cruttwell, 2014b].

Theorem 18.4 (Tangent descent)**.** *If T preserves the finite products defining $M_{\mathcal{U}}$ and $O_{\mathcal{U}}$, and preserves the equalizer defining $Z_{\mathcal{U}}$, then*

$$TZ_{\mathcal{U}} \cong \mathrm{Eq}(Td_0, Td_1).$$

Consequently, at a compatible smooth family z, an infinitesimal local update v preserves compatibility exactly when

$$T_z d_0(v) = T_z d_1(v).$$

Proof. Apply T to the equalizer cone. The preservation hypotheses make the lifted cone an equalizer. □

This is a preservation theorem. Away from the descent locus, tangent vectors to two restrictions lie over different base points and cannot be directly identified. Repair needs an additional lift.

18.5 Transverse repair

Suppose compatibility is represented locally by a regular constraint $g : M \to \mathbb{R}^m$, with descent locus $Z = g^{-1}(0)$. Given a predictive direction u_x, choose a_x so that

$$Dg_x a_x = -g(x) - Dg_x u_x,$$

then retract $x^+ = \mathrm{Ret}_x(u_x + a_x)$.

Proposition 18.5 (First-order repair)**.** *If g is twice continuously differentiable, Dg_x has a locally bounded right inverse, and the retraction is first-order accurate, then*

$$g(x^+) = O\Big(\|u_x + a_x\|^2\Big).$$

The linear compatibility defect is removed without descending $\|g(x)\|^2$.

Proof. Taylor expansion cancels the base and linear terms by the repair equation. □

If the linear solve has residual at most ε_x, the same argument gives

$$\|g(x^+)\| \le \varepsilon_x + c\|u_x + a_x\|^2.$$

Solver error, statistical uncertainty, and linearization error therefore remain separately auditable.

Boundary

Exact categorical notation does not make an estimated restriction exact. A repair is admitted only when its reduction exceeds uncertainty in the overlap witness and the predicted second-order remainder.

18.6 Restricted repairs and integrability

Let $\mathcal{A} \subset TM$ be the edits actually exposed by the architecture and set

$$\mathcal{D}_{\mathrm{SID}} = \mathcal{A}|_Z \cap TZ.$$

Theorem 18.6 (Admissible integrability)**.** *If $\mathcal{D}_{\mathrm{SID}}$ has constant rank and is closed under Lie brackets, then each compatible family lies on a maximal local integral manifold whose tangent directions are precisely the available gluing-preserving repairs.*

Proof. This is the Frobenius theorem applied to the restricted repair distribution. □

Bracket closure is informative only for a restricted repair language. If $\mathcal{A} = TM$, then $\mathcal{D}_{\mathrm{SID}} = TZ$ and closure is automatic. The theorem is local integrability, not convergence and not a universal argument for bracket regularization.

18.7 The SID controller

Sheaf Infinitesimal Descent

1: Estimate local predictive updates u_i
2: Evaluate typed compatibility and effectivity witnesses
3: Localize each witness to its incident cover simplices
4: Solve for admitted tangent corrections a_i
5: Retract $x_i \leftarrow \mathrm{Ret}_{x_i}(u_i + a_i)$
6: Audit pair, triple-overlap, and global-realizability conditions
7: Promote, damp, quarantine, collect data, or revise the cover

Method	Primitive	Structural consequence
Glue loss	scalar tradeoff	approximate pair agreement
Augmented Lagrangian	equality plus duals	solver-dependent feasibility
Projection	realized constraint	direct local feasibility
SID	descent sketch and repair lift	typed compatibility, effectivity, localization, and admission

Table 18.1: Similar linear algebra can implement different learning contracts.

18.8 Four foundry proofs of concept

Foundry	SID result	Preserved outcome	Claim boundary
Trade corridor	overlap residual 6.64×10^{-17} after one tangent lift	global RMSE 0.0918; promotion 0.966	synthetic nonmilitary predictive system
GLP-1 replay	residual 0, versus 0.0877 independent and 0.00202 glue loss	perfect global field and promotion accuracy	structural consistency, not medical truth
10-K replay	residual 0, versus 0.0290 independent and 0.000712 glue loss	perfect global field and promotion accuracy	frozen 46-claim audit, not financial advice
Product-feedback replay	exact compatibility; 0.976 witness retention; full context coverage	perfect contract and promotion accuracy	aggregate-interface audit, not review truth

Table 18.2: Thirty-seed proofs of concept retain prediction and promotion metrics separately from descent residuals.

Experiment: SID frozen-foundry replays

These are thirty-seed controlled replays from frozen local representations; reference labels score the run but never enter the correction. The GLP-1, 10-K, and product rows test descent plumbing rather than domain truth. The trade-corridor benchmark is particularly diagnostic: a tuned gluing penalty slightly improves local prediction over SID but leaves a nonzero witness, while a robust median improves global prediction without repairing compatibility. Aggregation, prediction, and descent are different operations.

Admission contract

A SID repair is admitted only when its typed witness is identifiable, the targeted compatibility or effectivity obstruction decreases beyond solver and statistical uncertainty, shared corrections agree across the cover, declared predictive and promotion safeguards pass, and no new obstruction appears. Otherwise the system damps, quarantines, gathers evidence, or revises the cover.

Design lesson

SID turns local-to-global structure into a learning interface. The cover localizes responsibility; the equalizer distinguishes compatibility from effectivity; tangent descent preserves admissible families; transverse repair returns obstructed families toward them; and admission prevents numerical agreement from being mistaken for global truth. This is the sheaf-theoretic core behind foundry persistence.

Further reading

The classical background for sheaves, descent, and topos semantics is given by Mac Lane and Moerdijk [1992]. Predictive state representations originate in early work on predictive representations and dynamical systems [Littman et al., 2001, Singh et al., 2004b]; later work connects prediction to planning and abstraction [Boots et al., 2011, Soni and Singh, 2007]. These readings clarify the two ingredients SID combines: local-to-global consistency and state represented by observable predictions.

Machine-learning uses of sheaves include sheaf neural networks [Hansen and Gebhart, 2020], connection-Laplacian models [Barbero et al., 2022], neural sheaf diffusion [Bodnar et al., 2022a], and nonlinear adaptive sheaf diffusion [Zaghen et al., 2024]. Copresheaf topological neural networks extend the architecture-level viewpoint [Hajij et al.,

2025]. SID addresses a different layer: it treats the declared descent diagram as a constraint and repair interface, including effectivity and admission, rather than choosing a sheaf Laplacian as the predictive architecture.

19
Infinitesimal Argument Repair

SCOLT

The Sheaf Chain of Toulmin Thought, or SCoTT, represents an argument as local Toulmin tickets and asks whether their typed roles glue. Chapter 10 introduced CoLT as a general architecture for reasoning by structural repair. Its Toulmin specialization is SCoLT, the Sheaf-theoretic Chain of LINCS Thought. SCoLT keeps the SCoTT base construction fixed and asks a further question:

> If an admissible argument is perturbed, transported, composed, or repaired, does it remain warranted, and can a failure be localized and corrected without changing its declared meaning?

The distinction produces a simple hierarchy. Chain of Thought generates a reasoning sequence. Chain of Toulmin Thought assigns argument roles. SCoTT tests whether local Toulmin structure glues. CoLT supplies the general diagnose–repair–admit architecture, and SCoLT differentiates, localizes, and safely repairs a failure in the Toulmin argument sheaf.

SCoTT supplies the base argument sheaf. SCoLT is its governed CoLT evolution under perturbation, localization, repair, and admission.

Boundary

SCoTT provides the static sheaf-theoretic criterion: local Toulmin tickets must agree on overlaps and admit a global realization. SCoLT adds a dynamical criterion: that agreement must remain stable under declared perturbations, and any repair must preserve the argument's licensed meaning. The deterministic E_0 study isolates this architectural claim; it does not assess unrestricted natural-language reasoning.

19.1 The Toulmin learning sketch

Let

$$\mathbb{S}_{\mathrm{Arg}} = (S_{\mathrm{Arg}}, \mathcal{D}_{\mathrm{Arg}}, \mathcal{L}_{\mathrm{Arg}}, \mathcal{K}_{\mathrm{Arg}})$$

be the argument learning sketch. Its objects include claims C, grounds G, warrants W, backing B, qualifiers Q, rebuttals R, source-bearing proposition clusters, local tickets, compatible families, and admitted global arguments.

Its arrows perform role extraction, source pullback, restriction, warrant licensing, backing support, qualifier scoping, rebuttal discharge, graph incidence, local-to-global realization, and projection to a downstream decision.

The declaration contains four particularly important path obligations.

Warrant transport. Restricting a warranted argument must agree with restricting its grounds, warrant, and claim before reconstructing the local bridge:

$$\rho_C(W(G)) \cong \rho_W(W)(\rho_G(G)).$$

Source naturality. A transported claim must retain the source restriction that licensed the grounds. Source erasure is a path failure even if the surface claim is unchanged.

Qualifier and rebuttal preservation. Transport may tighten a qualifier but cannot silently strengthen the claim. A rebuttal must remain explicit, be discharged by a declared evidence-bearing rule, or block promotion.

Descent and effectivity. Compatible tickets form a cone over their overlap diagram. A distinct realization map determines whether that compatible family is an admissible global argument.

19.2 Base and tangent argument obstructions

For a realized argument diagram

$$D_{\mathrm{Arg}} : J_{\mathrm{Arg}} \longrightarrow \mathcal{C},$$

the base obstruction is

$$O_0 = \mathcal{O}_{\mathrm{S}_{\mathrm{Arg}}}(D_{\mathrm{Arg}}).$$

It represents admissible diagnostics of missing or incompatible claims, grounds, warrants, qualifiers, rebuttals, sources, graph obligations, and global promotion.

Choose a declared tangent site $\mathcal{T}_{\mathrm{Arg}}$ and lift the same sketch:

$$O_1 = \mathcal{O}_{\mathrm{S}_{\mathrm{Arg}}}(TD_{\mathrm{Arg}}).$$

The tangent obstruction asks whether warrant structure survives declared changes in wording, evidence, source, context, model, prompt, intervention regime, and preference population.

Tangent site	Example probe	Characteristic failure
Surface form	Meaning-preserving paraphrase	Parser- or wording-dependent role assignment
Evidence	Delete or substitute one ground	Unsupported claim or discontinuous warrant
Warrant	Swap the inference rule while holding roles fixed	False bridge acceptance
Qualifier	Tighten or relax scope	Silent strengthening of the claim
Rebuttal	Inject, remove, or strengthen an exception	Obstruction erasure
Source	Move evidence across provenance domains	Source pullback failure
Context	Restrict or expand the local chart	Failure of naturality under context change
Preference population	Change annotator or prompt stratum	Hidden cyclic or heterogeneous preference obstruction

Table 19.1: A tangent site declares which argument variations carry semantic meaning.

19.3 *Quotient, localization, and repair*

Not every textual difference should trigger repair. A task-specific quotient

$$\chi : (O_0, O_1) \longrightarrow (\bar{O}_0, \bar{O}_1)$$

removes decision-null variation. Meaning-preserving formatting, reordering of independent grounds, and shared stylistic changes may be null. Qualifier, source, warrant, and rebuttal changes are not null when they can affect promotion.

The remaining obstruction is localized over a cover of reasoning steps, source passages, retrieved documents, proposition clusters, warrant bridges, model runs, populations, and downstream decision branches. The output is a compatible family of typed witnesses rather than an unordered collection of scores.

Repair has two modes. Tangential preservation changes presentation while remaining in the same admitted semantic family. Transverse repair changes a structurally material component: restoring evidence, weakening an overstatement, reinstating a rebuttal, replacing a warrant, or requesting missing support.

Admission contract

A Toulmin repair must remove the targeted paired obstruction, introduce no new typed obstruction, preserve source binding, respect the decision quotient, and pass held-out probes. If the cover is incomplete or the repair cannot be identified, the correct outcome is abstention or evidence collection.

19.4 E_0: deterministic structural perturbations

The E_0 study asks whether paired SCoLT profiles distinguish declared null changes from material changes to corpus-derived tickets. The evaluation uses all tickets produced from the 340-document Web-discourse corpus of Habernal and Gurevych and the 112 English argumentative microtexts of Peldszus and Stede, with no fitted parameters and therefore no train/test split.

Null controls changed formatting or reversed independent grounds. Material probes dropped a claim, warrant, ground, or rebuttal; strengthened a qualifier; or swapped a warrant or source binding. Each material probe changed one declared obligation. Inapplicable probes were omitted rather than counted as negatives; swaps used the next non-equivalent populated ticket within the same corpus.

The one-point comparator inspected only the perturbed ticket and detected missing required fields. The paired LINCS profile compared anchor and perturbation after applying the quotient, then emitted typed changes for claim, grounds, warrant, qualifier, rebuttal, and source.

Measure	One-point base gate	SCoLT
Sensitivity to material change	0.4090	1.0000
Specificity on null controls	0.9783	1.0000
Balanced accuracy	0.6936	1.0000
F1	0.5786	1.0000
Exact obstruction localization	not available	1.0000

Table 19.2: E_0 results on 564 corpus-derived tickets and 4,283 perturbation pairs.

The observed balanced-accuracy improvement was 0.3064, exceeding the pre-specified 0.25 criterion. The strongest base-gate blind spots were populated-but-swapped warrants, changed source bindings, strengthened qualifiers, and removed rebuttals.

Repair controls made the structural distinction sharper. LINCS admitted every exact restoration and rejected every generic nonempty surface patch. The base completeness gate admitted 96.99% of those superficial patches.

Experiment: what E_0 validates

E_0 validates the executable distinction between checking whether a ticket is populated and checking whether its typed obligations remain invariant under a declared change. It also validates the two quotient-null controls, typed localization, and blockwise rejection of nonempty but unwarranted repairs.

Boundary

The perturbations and evaluator share a typed structured representation. The perfect LINCS result therefore validates factorization, quotient, localization, and admission plumbing. It does not demonstrate robustness to natural paraphrases, noisy extraction, learned repair, answer correctness, automatic-differentiation semantics for language, or preference alignment.

19.5 *Limits and evidential ladder*

E_0 establishes that the typed machinery behaves correctly when perturbations and tickets share a structured representation. Stronger claims require three additional empirical regimes.

1. *Extraction robustness* requires held-out human or model-generated paraphrases and revisions evaluated after imperfect Toulmin extraction.
2. *End-to-end repair* requires language-model proposals compared under guarded admission, scalar reranking, unconstrained self-revision, and abstention.
3. *Preference-sensitive admission* requires testing whether preferences among argument candidates admit a scalar reward, with a relational fallback when cycles or population heterogeneity remain.

The present evidence therefore supports a narrow but substantive conclusion. SCoLT validates typed perturbation, quotienting, localization, and guarded repair for structured argument tickets. It does not yet establish robust extraction from unrestricted prose, reliable generation of novel repairs, or superiority to scalar baselines in end-to-end argumentation.

Further reading

The conceptual starting point is Toulmin's original account of claims, grounds, warrants, backing, qualifiers, and rebuttals [Toulmin, 1958].

The Argumentative Microtext corpus [Peldszus and Stede, 2015] and Web-discourse argument mining [Habernal and Gurevych, 2017] provide useful empirical settings in which argument components and relations are explicitly annotated.

For modern language models, zero-shot Toulmin explication [Gupta et al., 2024] shows that named argument roles can guide structured extraction, while fine-tuned LLM argument mining [Cabessa et al., 2025] represents the broader end-to-end extraction literature. TRACE [Kim and Yang, 2026] uses Toulmin elements to assess chain-of-thought reasoning. SCoLT is adjacent to all three but asks a different question: after roles have been extracted, do warrants, sources, qualifiers, and rebuttals restrict and glue coherently, and can a proposed repair be admitted without silently changing the claim?

20

Trustworthy Foundation Models

ODYSSEY AND PROMETHEUS

Foundation models are usually identified with the parameters of a large pretrained network. That identification is too narrow for systems expected to support research, decisions, and persistent institutional knowledge. The model may generate a fluent candidate answer, but the answer also depends on sources, retrieval, tools, intermediate artifacts, domain assumptions, cross-document joins, and rules governing what may be retained. Trust therefore cannot be a property of the neural model alone.

Kimi K3 makes this boundary concrete. Releasing a 2.8-trillion-parameter open-weight model does not by itself make the model operational: sparse expert routing, hybrid long-context attention, low-precision realization, inference kernels, and distributed execution are part of the usable system [Kimi Team, 2026]. Chapter 12 treats these interacting layers as a frontier-scale example of architectural obligations and repair. The weights are indispensable, but they are not the whole maintained artifact.

This chapter develops a complementary view. A trustworthy foundation model is a *maintained foundry*: a typed system of local representations whose claims can be traced to evidence, compared on overlaps, repaired when incompatible, and admitted into durable state only under an explicit contract. Odyssey supplies the governance and admission architecture for such foundries. Prometheus supplies a deep-research engine that constructs and inspects candidate world-model artifacts. Their division of labor realizes the LINCS workflow at system scale [Mahadevan, 2026j,k].

In this setting the distinction between a model of the world and a model of learning becomes operational. Prometheus may construct a biological, financial, or scientific world model whose mechanisms support domain questions. Odyssey models the process by which such artifacts are sourced, joined, challenged, revised, and promoted. LINCS makes that second-order system actionable: one can intervene

on an extractor, cover, argument rule, refresh policy, or admission block while preserving the unaffected foundry obligations. Such an edit is structural surgery on the learning system, not automatically a causal intervention in the represented world.

A generated artifact is a proposal. It becomes maintained knowledge only after its provenance, restrictions, overlaps, arguments, and intended use have survived independent admission.

The phrase *truth-preserving* is deliberately local. It means preserving declared evidence and compatibility conditions under transport. It does not mean that the system possesses an oracle for external truth.

Boundary

The architecture addresses structural trustworthiness: provenance loss, scope drift, incompatible local claims, unsupported argument transport, stale artifacts, and unsafe promotion. It cannot make an unreliable source reliable, prove that a cover is complete, or turn a plausible causal claim into an identified causal effect. Those remain evidence and validation problems.

20.1 *From a model to a foundry*

The term *foundation model* emphasizes broad reuse across downstream tasks [Bommasani et al., 2021]. The foundry view asks what must surround that reusable substrate before its outputs can support accountable work. A convenient declaration is

$$\mathfrak{F} = (\mathbb{S}, \mathcal{U}, \mathcal{E}, \rho, \mathcal{O}, \mathcal{A}, \mathcal{V}).$$

Here $\mathbb{S}$ is the sketch of objects and required paths; $\mathcal{U} = \{U_i \to U\}$ is the localization cover; $\mathcal{E}$ is the evidence-bearing presheaf or sheaf; ρ denotes restriction and transport maps; $\mathcal{O}$ is the obstruction ledger; $\mathcal{A}$ is the admission contract; and $\mathcal{V}$ is the versioned, refreshable state that has already been admitted.

This declaration changes the unit of trust. The unit is not an entire model and not an unqualified sentence. It is a typed claim in a local context, connected to a source span and an argument, with a recorded status under a particular admission contract. Different portions of one generated report may consequently have different statuses.

Definition 20.1 (Locally trustworthy foundry state)**.** An artifact x is locally trustworthy relative to a foundry declaration $\mathfrak{F}$ when its evidence restrictions are defined, its declared overlaps satisfy the applicable compatibility laws, its argument ticket licenses the intended conclusion, and its admission blocks accept it for a specified use. The definition is relative to $\mathfrak{F}$, its sources, and its audit family; it is not a claim of absolute truth.

Four separations follow.

Generation is separated from admission. The process that proposes an artifact should not be the sole authority that promotes it.

Compatibility is separated from effectivity. Local claims can agree pairwise without supporting a usable global object. The system must check both overlap agreement and the existence of an admissible realization.

Evidence is separated from argument. A source span can support grounds without licensing the warrant used to reach the conclusion. Toulmin structure makes that gap inspectable.

Current state is separated from persistent state. A fresh run may be queryable while remaining quarantined. Durable state requires an explicit update and refresh decision.

20.2 *The Odyssey–Prometheus division of labor*

Odyssey and Prometheus occupy different positions in the architecture. Prometheus acquires heterogeneous sources, extracts structured events and claims, builds local predictive or causal representations, checks overlaps, and emits an artifact bundle. Odyssey declares what is being built, governs the workflow, validates representation laws, exposes arguments, and decides what becomes maintained foundry state.

Role	Primary responsibility	Trust question
Scylla	Human-facing model brief and answer contract	What is being claimed, for whom, and under what scope?
Homer	Workflow orchestration, checkpoints, and replay	Can the construction be reproduced and refreshed?
Athena	Cover, local types, restrictions, and gluing laws	Do the representations compose as declared?
Prometheus	Source acquisition and candidate world-model artifacts	What local evidence, models, and tensions can be constructed?
Toulmin	Claim, grounds, warrant, backing, qualifier, and rebuttal	Does the evidence license the stated conclusion?
TICKET	Promotion, qualification, quarantine, or rejection	May this artifact enter maintained state for the requested use?

Table 20.1: The roles separate proposal, structural validation, argument scrutiny, and admission.

The names are less important than the separation of powers. A Prometheus bundle can contain a source manifest, artifact manifest, world-model description, restriction and gluing audit, obstruction ledger, dashboards, and refresh plan. These artifacts remain inspectable even if Odyssey declines to promote them. Conversely, Odyssey can export an admitted foundry slice to Prometheus for exploration without surrendering the admission boundary.

Design principle

The artifact producer should expose enough structure for another component to reject, qualify, or replay its work. Trustworthy generation requires a first-class refusal path.

20.3 Safety as an admission problem

Many safety mechanisms operate on the final response: filter a prohibited request, score toxicity, or inspect a completed answer. Those mechanisms are necessary in suitable settings, but a research foundry has a longer failure surface. A source may be silently dropped; a claim may migrate to a broader population; two reports may use the same word for different variables; a numeric statement may lose its units or period; a rebuttal may disappear during summarization; or a provisional inference may be stored as fact.

LINCS treats these as failures of declared compositionality. For a candidate artifact x, an admission record may retain a structured residual

$$\Omega(x) = \big(\Omega_{\text{source}}, \Omega_{\text{scope}}, \Omega_{\text{overlap}}, \Omega_{\text{argument}}, \Omega_{\text{freshness}}, \Omega_{\text{use}}\big).$$

The components are not collapsed prematurely into a single safety score. They identify different repairs and confer different authority. A missing source requires retrieval or abstention; a scope mismatch requires qualification; an overlap tension may require a split cover; an unsupported warrant requires new grounds or a weaker conclusion; and stale evidence requires refresh.

The admission map is correspondingly typed:

$$\mathrm{Admit}_{\mathfrak{F}}(x) \in \{\mathsf{promote}, \mathsf{qualify}, \mathsf{quarantine}, \mathsf{reject}, \mathsf{abstain}\}.$$

These outcomes should not be read as a total ranking. A quarantined artifact may be useful for inspection but ineligible for downstream decisions. A qualified claim may be admissible in one population and forbidden in another. Abstention records that the declaration does not license an answer.

Admission contract

No generated claim enters durable foundry state merely because it is fluent, probable, or internally consistent. Promotion requires source-bearing local sections, declared restriction maps, a retained obstruction record, an argument ticket, a use-specific gate, and a refresh policy.

Chain-of-Evidence offers a complementary operational precedent: provenance is maintained while an autonomous research artifact is constructed, and the resulting claims are checked against literature, code, evaluator outputs, and experiment logs [Meng et al., 2026]. A LINCS foundry places those evidence links inside a broader admission state that also retains scope, local-to-global compatibility, rejected repairs, and refresh obligations.

20.4 *The six-stage workflow at foundry scale*

The architecture is an instance of the six-stage workflow developed in Chapter 2. The correspondence is especially useful because it distinguishes model improvement from the governance of model state.

Stage	Foundry operation	Typical safety artifact
Declare	Specify task, sketch, cover, sources, roles, and intended use	Scylla brief and Athena plan
Differentiate	Probe source removal, paraphrase, regime change, or model variation	Sensitivity and counterfactual audit
Quotient	Identify duplicate, presentation-equivalent, or decision-null variation	Canonical source and claim identities
Localize	Assign failure to evidence, overlap, argument, or workflow chart	Obstruction ledger with witnesses
Repair	Retrieve, split, qualify, re-run, or revise the warrant	Versioned repair and replay trace
Admit	Apply structural, evidential, semantic, and operational gates	TICKET decision and refresh obligation

Table 20.2: LINCS operations become governance operations when the learned object is a maintained foundation-model foundry.

Infinitesimal probes have a practical role here. Removing one source, perturbing one premise, changing a reporting period, or transporting a claim to an adjacent population asks whether the admitted conclusion is locally stable. Large discontinuities remain possible: a new ontology, causal mechanism, or foundry cover may have to be proposed. As

Chapter 22 explains, the infinitesimal layer diagnoses pressure on the current declaration; it need not pretend that every architectural repair is a small parameter update.

20.5 Three foundry case studies

The same architecture becomes concrete in very different domains. Health research stresses population, intervention, and safety qualifiers. Corporate filings stress source anchors, units, periods, and narrative–numeric agreement. Product research stresses identity, review heterogeneity, and transport between consumer, store, brand, and corporate contexts.

The evidential unit in these studies is an artifact record, not a generated sentence scored in isolation. Each record carries a source identifier and span, local type and scope, restriction outcomes, argument status, admission decision, and refresh obligation. Counts below are descriptive audit totals from frozen foundry runs. They establish that the architecture can preserve and expose these distinctions; they are not estimates of population-level answer accuracy, causal correctness, or professional decision quality.

20.5.1 A GLP-1 health-research atlas

The Prometheus GLP-1 study retained eleven heterogeneous documents rather than forcing them into a single evidential genre. The corpus included efficacy studies, safety and adverse-event material, access and adherence, cardiometabolic outcomes, delivery technologies, and adjacent mechanistic work. Its construction produced 3,376 extracted events, 11 causal episodes, 191 local predictive-state contexts, 190 restriction checks, 186 compatible gluing overlaps, and four tense overlaps. Of the restrictions, 149 were compatible and 41 divergent.

Those numbers describe an atlas, not a vote. Evidence for core weight-loss efficacy can glue across suitable clinical contexts while gastrointestinal mediation, hair loss, hormonal effects, reward pathways, and adjacent animal mechanisms remain local, qualified, or obstructed. The appropriate output is therefore not the global sentence "GLP-1 works." It is a navigable collection of claims whose intervention, population, outcome, evidence level, and rebuttals remain visible.

In a health foundry, preserving a contradiction can be safer than resolving it. Two studies may differ because their populations, interventions, outcomes, or evidence levels should never have been glued.

This case illustrates why non-compositionality is not merely another loss. Each divergent restriction carries a type and a proposed next observation. Scalar averaging would erase the distinction between a genuine contradiction and a legitimate regime boundary.

20.5.2 A corporate 10-K foundry

A corporation foundry uses a cover over financial statements, risk factors, management discussion, strategy signals, and market context. The overlaps encode questions such as whether financial trends agree with the narrative, whether strategy claims survive risk-factor rebuttals, and whether a claimed market advantage is supported outside the filing.

One Adobe 10-K workflow produced 46 candidate claims: 28 were supported and 18 were retained with qualifications. Its gates required filing-local source anchors, exact table–unit–period checks for numeric claims, risk-factor rebuttals for strategic synthesis, and preservation of upstream identifiers and qualifiers across role handoffs. The result was marked ready for Scylla review rather than declared globally true.

A separate corporation gluing audit illustrates the value of retained failure. Financial–risk, financial–narrative, and narrative–strategy overlaps glued, while the strategy–market overlap remained obstructed pending further filing passages and market evidence. A conventional summary could hide that asymmetry inside a polished account. The foundry instead keeps the unsupported moat inference out of trusted state.

20.5.3 Product and review foundries

A product-review foundry declares restriction ports for product identity, review corpus, rating metadata, theme surface, claim evidence, and negative-review rebuttals. A broader product foundry can then join those charts to store experience, assortment, brand promise, and corporate context.

This decomposition prevents three common collapses. Reviews for adjacent products are not evidence for the focal product merely because their names are similar. A frequent review theme is not automatically a population-level fact. And a brand promise cannot be inferred from consumer sentiment without a declared bridge to corporate evidence. Product identity, source provenance, corpus coverage, and negative evidence remain first-class obligations.

Experiment: frozen trustworthy-foundry audits

The GLP-1 atlas records 3,376 extracted events, 191 local predictive-state contexts, and 190 restriction checks; the Adobe 10-K workflow records 46 candidate claims, of which 28 are supported and 18 retained with qualifications. These are descriptive artifact audits demonstrating retained provenance, scope, and obstruction states. They are not population-level estimates of medical, financial, or product-answer correctness.

Table 20.3: Trustworthiness depends on the domain-specific cover and admission law, not on a universal confidence threshold.

Foundry	Unsafe global collapse	Localizing cover	Characteristic gate
GLP-1 research	One verdict across trials, safety reports, populations, and mechanisms	Population, intervention, outcome, evidence level	Regime-compatible clinical claim with explicit qualifier
Corporate 10-K	Fluent strategy narrative detached from numbers and risks	Statement, risk, management, strategy, market	Source anchor plus exact unit, period, and rebuttal
Product research	Aggregate sentiment transported across products, stores, and brand claims	Identity, reviews, ratings, themes, store, brand	Product-local evidence and retained negative-review rebuttal

20.6 A compositional account of AI safety

The foundry perspective does not replace alignment, robustness, privacy, security, or human-factors research. It contributes a structural layer that connects them. A safety requirement becomes operational when one can say which diagram should commute, which perturbations are meaningful, which variation is irrelevant, where failure can be localized, and what repair may be admitted.

Five recurring safety failures acquire compositional forms:

1. *provenance loss*: source restriction no longer commutes with claim transport;
2. *scope drift*: a claim is pushed into a population or use not covered by its qualifier;

3. *context collapse*: locally valid sections are glued across an obstructed overlap;
4. *argument laundering*: grounds survive summarization while the warrant, rebuttal, or uncertainty disappears; and
5. *state contamination*: a candidate or stale artifact is treated as admitted current knowledge.

These failures are related but not interchangeable. Their obstruction types determine whether the correct response is retrieval, qualification, cover refinement, argument repair, refresh, quarantine, or abstention. The system learns from the repair traces as well: repeated gate failures reveal weak workflow skills, incomplete covers, and brittle transports. Policy optimization can then improve the proposal and audit process while the admission boundary remains fixed.

20.7 What the architecture does not guarantee

Structural auditability is valuable only if its limits remain explicit.

External truth. Agreement among sources may reproduce a shared error. Source manifests and gluing cannot substitute for independent measurement or domain expertise.

Cover completeness. An omitted population, risk class, reporting period, or stakeholder may make all recorded overlaps look compatible. Cover design is itself a scientific and governance claim.

Causal identification. A coherent causal story is not an identified effect. Interventional and latent-confounding assumptions still require the diagnostics developed in Chapters 5 and 11.

Audit independence. Agents built from the same model family may share blind spots. Deterministic checks, heterogeneous models, external tools, and human review provide different forms of independence, none of them automatic.

Adversarial robustness. Typed artifacts expose attack surfaces but do not eliminate prompt injection, poisoned sources, compromised tools, or malicious operators. Security must surround the foundry.

Normative legitimacy. An admission contract can make a policy explicit and auditable; it cannot by itself establish that the policy is fair, lawful, or socially legitimate.

Boundary

The strongest defensible claim is not "the architecture makes foundation models safe." It is that the architecture makes specific trust obligations, failures, repairs, and promotion decisions representable and inspectable. That is a precondition for serious safety work, not its completion.

20.8 Toward foundry-scale learning

Odyssey and Prometheus broaden the object optimized by LINCS. Earlier chapters repair a causal model, an adapter family, a skill policy, a reward relation, or an argument. A trustworthy foundation-model system must also repair the processes that create and maintain those objects. Its learned state includes covers, source policies, extraction skills, gluing tests, argument templates, admission rules, and refresh schedules.

This produces two coupled loops. The *inner loop* builds and repairs a domain foundry under a fixed declaration. The *outer loop* studies repeated obstruction and admission traces to revise the foundry declaration or the skills that realize it. The outer loop must not erase its own evidence: a proposed new cover or policy is itself a candidate object subject to comparison, replay, and admission.

The same structure points toward trustworthy scientific discovery. A Prometheus-style engine may propose a new mechanism or connection that does not fit the current sketch. Odyssey-style governance can preserve the anomaly that motivated the proposal, transport prior evidence into the enlarged declaration, and require a novel prediction or discriminating observation before promotion. This is not yet a general theory of creativity. It is, however, a way to keep abductive proposals connected to evidence without reducing novelty to a scalar reward.

The broader conclusion is that foundation-model trust cannot be obtained by making only the base neural predictor more capable. The surrounding learning process must itself become a maintained model with named components, intervention rights, invariants, and evidence-bearing admission. That is the foundry-scale form of the causal lesson: an intelligent system becomes responsibly actionable only to the extent that the mechanisms of its own knowledge construction are explicit enough to inspect and repair.

20.9 *Further reading*

The foundation-model literature surveys both the opportunities created by broadly reusable models and the risks introduced by homogenization, emergence, and downstream adaptation [Bommasani et al., 2021]. Prometheus develops the automated deep-research and world-model construction layer [Mahadevan, 2026k]. Odyssey develops the complementary foundry algebra, typed agent roles, artifact transport, and admission architecture [Mahadevan, 2026j]. Chapter 19 supplies the argument-repair layer used to distinguish evidence possession from warranted inference, while Chapter 21 abstracts the declaration, localization, repair, and admission choices into a reusable design method.

Part IV

Synthesis

21
A Pattern Language for LINCS Systems

The applications differ in their data, models, and repair mechanisms, but they share a stable set of design questions. This chapter turns those questions into a pattern language for constructing and auditing new LINCS systems.

21.1 The declaration card

Every system begins with a declaration card containing:

1. the learning sketch and its intended factorization;
2. the base and tangent obstruction objects;
3. the observation maps used for measurement;
4. the decision quotient;
5. the localization cover and its completeness claim;
6. the allowed repair language;
7. the admission blocks;
8. the abstention conditions;
9. the certificate emitted after a decision;
10. the structural level at which each repair acts; and
11. the non-target obligations that every repair must preserve.

Design principle

If an application cannot fill out the declaration card before optimization, its residual has not yet been given enough structure to function as a LINCS learning signal.

21.2 *Six recurring patterns*

Factorization before scalarization. State what must compose before choosing how failure will be measured.

Probe semantics before differentiation. Specify what a tangent direction means in the domain. Automatic differentiation is one implementation, not the definition of a meaningful probe.

Quotient before repair. Remove gauge, presentation, and decision-null variation before assigning repair effort.

Complete localization before blame. Local witnesses justify component-level diagnosis only relative to a cover that can reconstruct the declared global failure.

Typed repair before optimization. The correction space should encode what the system is allowed to change.

Blockwise admission before deployment. Structural improvement, semantic preservation, statistical support, and operational feasibility should remain separately auditable.

21.3 *The actionable-model pattern*

A LINCS declaration should make the learner actionable, not merely descriptive. For every proposed repair it should be possible to answer four questions:

1. Is the target a parameter, component, architecture, or declaration?
2. Which objects, arrows, observers, and semantic commitments are held fixed?
3. How does the local edit propagate to the behavior being evaluated?
4. Which evidence, not used to generate the proposal, can admit or reject the result?

This is the most general lesson LINCS takes from causal inference. Causal models answer these questions for domain mechanisms. LINCS asks them of the learning system itself. When the modeled arrows also carry causal semantics, the two levels can be connected; otherwise the result remains a structural intervention and should be described that way.

21.4 The cross-application map

System	Declared object	Obstruction	Quotient or cover	Repair contract
CoLT	Maintained reasoning state and transition record	Typed failure of a declared commitment	Reasoning contexts and decision-null variation	Certified admit, reject, or abstain
BRIDGE/SKFM	Intervention response fields	Visible bracket nonclosure	Regime, target, and spectral covers	Screen plus identified discovery
LINCS-KET	Direct and Kan routes	Route incompatibility	Layers, incidence, Cech probes	Blockwise post-training gate
ALLORA	Adapter fields	Order-sensitive composition	Symmetry and layer structure	Capability-safety preservation
LASKO	Skill sections	Anchor, closure, feasibility	Kernel/image fibers, workflows	Executable skill gate
GIRL	Bellman factorization	Base/tangent Bellman failure	Advantage quotient, computation cover	Layered policy recovery
LINCS-RLHF	Preference field	Cyclic or unsupported reward	Population cover, cycle space	Scalar gate or relational fallback
RADAR	Relational charts	Incidence, cocycle, apex failure	Private faces and shared joins	Sparse decision-preserving descent
SID	Predictive-state sheaf	Compatibility or effectivity	Cech cover and tangent equalizer	Tangential or transverse update
SCoLT	Argument sheaf	Warrant, source, scope, rebuttal	Role and source cover	Typed repair certificate
Odyssey/Prometheus	Maintained foundry	Provenance, overlap, argument, freshness	Evidence and domain cover	Typed admission or quarantine

21.5 A minimal experiment contract

A credible experiment separates four questions:

1. Does the registered probe actually change the declared obstruction?
2. Can the method localize that change without reacting to null controls?
3. Does the repair improve held-out structural audits?
4. Does the repaired system preserve the task semantics and operational constraints?

Retained failures are part of this contract. The LINCS-KET joint-norm result, for example, explains why blockwise admission exists. A negative experiment that identifies a missing gate contributes more to the pattern language than an unqualified reduction in a training residual.

Admission contract

No aggregate score may silently compensate for failure of a mandatory block. Weights are useful inside a declared block; they do not replace the logical structure of the admission decision.

21.6 *Certificates and reproducibility*

The output of a LINCS system is not only a repaired model. It is a replayable certificate containing the declaration, probes, quotient decisions, localization witnesses, proposed repair, admission results, residual obstructions, and abstention reason when applicable.

This certificate is the common interface across all nine applications. It also supplies the natural bridge from research prototypes to scientific and operational audit. It records not only what changed, but at which structural level the change occurred and which invariants were claimed to survive.

Further reading

The pattern language rests on several mature literatures rather than one single precursor. Sketches and categorical model theory provide declaration [Barr and Wells, 1999, Makkai and Paré, 1989]; tangent categories provide structural differentiation [Cockett and Cruttwell, 2014b]; sheaves provide localization and descent [Mac Lane and Moerdijk, 1992]; and categorical learning provides compositional parameter and credit semantics [Fong et al., 2019, Cruttwell et al., 2022].

For the empirical side, reproducible language-model evaluation requires fixed task definitions, versions, prompts, decoding rules, and complete call accounting. The Language Model Evaluation Harness [EleutherAI, 2026] is one widely used substrate, while recent lessons on reproducible evaluation emphasize the instability introduced by seemingly minor harness choices [Biderman et al., 2026]. A LINCS certificate adds the structural declaration, null quotient, localization witness, rejected repairs, and admission decision to this familiar experiment manifest.

22
Frontiers of Infinitesimal Learning

LINCS begins with a declared structure. The sketch says which compositions matter; the tangent site says which perturbations are meaningful; the quotient says which variations are null; the cover says where failure can be observed; and the admission contract says what evidence licenses repair. This separation has made the applications in this book comparable without making them interchangeable.

The frontier is to let some of this structure evolve. A learning system might discover a missing context, enlarge a repair language, refine an interaction signature, or replace strict equality by coherent deformation. But a system that learns its own test can also learn to hide its failure. The central research problem is therefore not unrestricted self-modification. It is *auditable structural adaptation*.

The next generation of LINCS systems must learn declarations and diagnostics without allowing repair to rewrite the criterion by which it is judged.

22.1 *Compositionality and compression*

One influential tradition identifies learning with compression. In its minimum-description-length form, a learner selects a hypothesis H by balancing the code for the hypothesis against the code for the data given the hypothesis:

$$\widehat{H}_{\mathrm{MDL}} \in \arg\min_{H} \{ L(H) + L(X \mid H) \}.$$

Algorithmic information theory idealizes this intuition through Kolmogorov complexity, while practical MDL makes the coding scheme and model class explicit [Rissanen, 1978, Grünwald, 2007, Li and Vitányi, 2019]. This is a powerful account of regularity and model selection. It is not, however, the primitive adopted by LINCS.

LINCS begins with a sketch $\mathbb{S}$, a realized diagram D, and the typed obstruction

$$\mathrm{Obs}(\mathrm{Fact}_{\mathbb{S}}(D)) .$$

The obstruction records which declared path, cone, cocone, restriction, or factorization failed. It need not assign a total order to candidate models, and it need not live in a numerical space. Description length

instead assigns a scalar after choosing a representation and code. The two constructions therefore operate at different logical levels: one diagnoses failure of a declared relationship; the other compares the economy of descriptions.

Neither property implies the other. A short program can generate data while systematically violating a safety constraint, a causal intervention law, or an argument's source restrictions. Such a program compresses well but remains non-compositional relative to the relevant sketch. In the other direction, a compositional system may need a long description because its certificate retains local evidence, provenance, uncertainty, rebuttals, or exceptional contexts. Those fields are not waste merely because deleting them shortens the code.

There is nevertheless a close and useful interaction. Shared morphisms, repeated diagrams, reusable local sections, and stable restriction maps can all support shorter descriptions. Compositionality can therefore be a source of compression. But even perfect compression of the observed sample does not identify which compositions ought to hold, which presentation changes are null, or which repair preserves the intended semantics. Those choices belong to the declaration, quotient, and admission contract.

This distinction also clarifies the role of coding dependence. Kolmogorov complexity is invariant only up to a machine-dependent additive constant and is not computable in general; operational MDL replaces it with an explicit code and model family. LINCS faces an analogous but typed presentation issue: before comparing lengths, it must state which encodings represent the same decision-relevant object. A quotient χ can remove declared presentation redundancy, after which a description-length functional

$$L_\chi(R) + L_\chi(X \mid R)$$

may compare candidate repairs R on the quotient. The quotient must precede the comparison; otherwise a coding convention can reward or punish variation that the application has already declared meaningless.

Design principle

Use compression to compare or regularize repairs within a declared admissibility class. Do not let a shorter code erase the structural promise, the obstruction witness, or the evidence required for admission.

The Kimi K3 case study in Chapter 12 makes the distinction operational. Latent expert representations, sparse activation, recurrent sequence state, block summaries over depth, and low-precision weights all reduce some computational or representational burden. Yet each

mechanism is admitted only relative to a different preservation question: routing balance, global retrieval, information flow across depth, task behavior, or train–serve consistency [Kimi Team, 2026]. Compression is pervasive in the system, but the architecture specifies what each compression is allowed to forget.

This suggests a genuine research program rather than a rivalry between frameworks. One can ask when code lengths are compatible with restriction and gluing, when local compression bounds assemble globally, whether an admitted repair is minimal among repairs of the same type, and how much description is required by the certificate itself. In this role, compression is an observer of a LINCS system—and sometimes a valuable search principle—but not its ontology of learning.

22.1.1 *The jump changes the declaration*

Zahavy's position paper *LLMs Can't Jump* advances a sharper challenge to creativity as compression [Zahavy, 2026]. Following Einstein's account of scientific discovery, it distinguishes induction, deduction, and abduction. Induction extracts a rule from cases and results; deduction derives results from fixed rules and cases; abduction proposes an explanatory case or rule for a surprising result. The paper interprets Einstein's transition from physical thought experiments to the axioms of general relativity as such a "jump."

The general-relativity example matters because the available predictive error was not commensurate with the conceptual reorganization eventually required. Newtonian gravity was empirically extremely successful, and the observations that later provided decisive tests of general relativity were sparse or subsequent to its formulation. A compression-driven search within the old hypothesis language could prefer a small patch to a change in the primitives of space, time, acceleration, and gravitation. The eventual theory was elegant, but its discovery path initially enlarged and disrupted the representational scheme rather than monotonically shortening a code.

In LINCS terms, this is the difference between changing a realization D inside a declaration $\mathfrak{L}$ and changing the declaration itself:

$$\mathrm{Obs}(\mathrm{Fact}_S(D)) \;\rightsquigarrow\; \big(\Phi : \mathfrak{L} \to \mathfrak{L}'\big) \;\longrightarrow\; \text{transport and admission.}$$

The first arrow is abductive and proposal-generating: it may introduce new objects, morphisms, axioms, probes, or a new quotient. The second is auditable: it asks whether old witnesses survive transport, whether the new declaration resolves the motivating obstruction, and whether it makes new predictions or supports interventions that pass independent tests.

This formulation also shows how a learning signal can exist without

a large statistical residual. Individually successful local theories may fail to glue; requirements inherited from different domains may not commute; or an apparently harmless assumption may block a desired extension. Such failures are structural tensions among declarations, not necessarily errors on a supervised sample. A new theory can respond by reorganizing the diagram rather than fitting the old diagram more closely.

The connection should not be overstated. A historical case study is not a proof that no LLM can perform abduction, and no result in this book establishes that LINCS can outperform compression-based learning on creative discovery. Nor does detecting non-compositionality uniquely determine a new declaration. Zahavy proposes physically consistent, interactive world models as a source of grounded counterfactual simulation. Such a system could generate candidate declaration changes; LINCS could then type the proposal, preserve its provenance, transport prior evidence, and separate invention from admission.

Boundary

LINCS does not yet mechanize the abductive jump. It provides a formal boundary between repair within a frame and repair of the frame, together with the obligations a proposed new frame must satisfy.

22.1.2 *Can infinitesimal learning jump?*

The title of this book's final chapter creates an apparent tension with the claim that scientific creativity may require a jump. If LINCS is *infinitesimal*, is it confined to small improvements inside a fixed conceptual neighborhood?

The answer depends on what kind of jump is at issue. An infinitesimal method can describe a finite change whenever local directions integrate along a path. At a coarse temporal or observational resolution, the endpoint can look abrupt even though the trajectory is locally resolved. Long-run evolutionary change sometimes has this character: accumulated variation, inheritance, selection, drift, and ecological interaction can produce a large phenotypic or population-level separation. This example is suggestive, not a universal reduction. Mutations and lineage events may themselves be discrete, and evolutionary dynamics can cross thresholds or reorganize developmental and ecological regimes.

Other jumps are *critical*. A small change moves a system through a bifurcation, phase boundary, or loss of stability, after which its qualitative organization differs. Infinitesimal analysis can reveal the approaching instability, null mode, curvature, or failure of closure. It does not make

the two regimes identical, and a first-order probe at a singular point may be insufficient.

The strongest jumps are *declarational*. A new generator, ontology, mechanism, or dual description may inhabit no smooth path inside the current model category. Quantization is not merely a small parameter update to a classical declaration; a biological mechanism transported from population theory is not already a tangent vector in the old biological vocabulary; and a bulk–boundary duality relates two theories rather than moving infinitesimally inside either one. Such proposals belong to the outer category of declarations and correspondences developed below.

Kind of apparent jump	Role of infinitesimal structure	Additional requirement
Integrated change	Supplies local directions and transports obstruction information along a path	Existence and admission of a finite integrated repair
Critical transition	Detects instability, interaction, or loss of closure near a regime boundary	Higher-order probes and validation on both sides of the transition
Declaration change	Localizes persistent pressure that the present language cannot absorb	A discrete proposal, transport of prior evidence, and independent admission

Table 22.1: Infinitesimal diagnosis supports several kinds of change without reducing every jump to a smooth trajectory.

This yields the appropriate claim for LINCS: infinitesimal diagnosis is valuable without being generatively complete. It provides typed local evidence, separates meaningful from null directions, and constrains where a larger repair must act. A proposal mechanism may then integrate those directions, cross a detected boundary, or make a discrete meta-repair. The admission mechanism must determine whether the resulting finite or frame-changing proposal actually explains the obstruction.

Design principle

Use infinitesimal structure as a microscope and a transport calculus, not as an ontology asserting that all novelty is continuous. A jump that leaves the current model space must remain visible as a change of declaration.

22.1.3 *Extending the workflow from repair to discovery*

The six-stage workflow in Chapter 2 is deliberately conservative. During an ordinary run, the declaration fixes what counts as a failure, which perturbations carry meaning, which variations are null, and which repairs may be admitted. Scientific discovery requires a controlled escape from that fixed frame without abandoning the discipline of independent admission.

Two complementary events can initiate such an extension. In the first, *persistent anomaly* reveals that no repair in the declared language reconciles theory and observation. The obstruction then points beyond a realization D toward a proposed change $\Phi : \mathfrak{L} \to \mathfrak{L}'$. In the second, a proposed declaration $\mathfrak{L}'$ generates a *novel consequence*: an observation not used to construct the proposal, but specific enough to discriminate it from available alternatives.

These are not merely two additional boxes appended to the workflow. They form an outer discovery loop around it:

$$\begin{aligned}&\mathsf{declare} \to \mathsf{differentiate} \to \mathsf{quotient} \to \mathsf{localize}\\ &\quad\to \begin{cases} \mathsf{repair}_{\mathfrak{L}}(D), & \text{within-frame learning,}\\ \Phi : \mathfrak{L} \to \mathfrak{L}', & \text{abductive frame revision,}\end{cases}\\ &\quad\to \mathsf{predict}(\mathfrak{L}') \to \mathsf{observe} \to \mathsf{admit,\ reject,\ or\ abstain.}\end{aligned}$$

Every candidate $\mathfrak{L}'$ begins a new six-stage run. It must declare its own paths and probes, expose rather than erase the obstruction that motivated it, and face evidence not controlled by the mechanism that proposed it.

Planck's black-body analysis illustrates the anomaly-driven direction. New spectral measurements showed that the Wien distribution law lacked general validity. Planck explicitly concluded that the theory required improvement and sought the replaceable link in its derivation. His revised statistical argument introduced finite energy elements with size proportional to frequency, $\varepsilon = h\nu$, and yielded a radiation law consistent with the measured spectrum [Planck, 1901]. The historical claim should be kept precise: Planck's step discretized the energy elements used in the resonator calculation; the stronger ontology of freely propagating light quanta came later. For the present purpose, the important feature is the structural leap. A failed relation among spectrum, entropy, and classical derivation was not repaired by another parameter fit. One of the theory's constitutive assumptions was replaced.

General relativity illustrates the prediction-driven direction. The completed theory implied that starlight grazing the Sun should be deflected by approximately 1.75 arcseconds [Einstein, 1916]. A total solar eclipse made the relevant stars observable close to the Sun, turning

a theoretical consequence into an experimental design. Expeditions to Príncipe and Sobral observed the eclipse of 29 May 1919; Dyson, Eddington, and Davidson reported measurements consistent with the general-relativistic value [Dyson et al., 1920]. Einstein had derived an earlier, incomplete light-deflection value in 1911, so the novelty was not simply the idea that gravity bends light. The mature theory supplied the discriminating magnitude and thereby specified a test capable of separating declarations.

The two episodes expose a symmetry that ordinary optimization often hides. Data can obstruct a theory strongly enough to demand a new primitive, while a new primitive can generate observations that did not previously belong to the experimental agenda. Scientific creativity moves in both directions:

$$\begin{aligned} \text{observation} &\longrightarrow \text{new declaration}, \\ \text{new declaration} &\longrightarrow \text{new observation}. \end{aligned}$$

LINCS can represent the first arrow as obstruction-driven proposal and the second as probe generation. Neither arrow is self-certifying.

Admission contract

An abductive declaration change must preserve an explicit witness of the anomaly that motivated it, explain why admissible within-frame repairs are insufficient, transport prior successes conservatively, generate discriminating consequences, and expose those consequences to independent observation. Novelty without such obligations is unconstrained invention; fit without novel consequences is only retrospective accommodation.

22.1.4 *Generative closure and functorial leaps*

Lie-bracket nonclosure in Chapter 11 provides a concrete model of how structural failure can initiate discovery. The observed response fields generate a bracket that lies outside their visible span. One possible repair adds a latent direction; others add an observed coordinate, introduce memory, separate regimes, or reject the field model. The residual does not name the missing object, but it constrains the family of declarations that could explain the failure.

Lie closure is too specialized to serve as a general theory of scientific creativity. Its broader analogue is *generative closure*. A declaration specifies operations under which its explanations are supposed to remain adequate. Applying those operations should not produce an unexplained object:

$$\mathcal{O}_{\mathrm{gen}}(D) = \mathrm{Gen}_{\mathfrak{L}}(D)/D.$$

The notation is schematic because the generator depends on the science. It may be a Lie bracket, logical consequence, symmetry action, conservation law, restriction map, intervention, composition, or experimental transport. Nonzero $\mathcal{O}_{\text{gen}}$ signals that the declared world is not closed under one of its own sanctioned constructions.

Scientific discovery also proceeds through a different operation: transporting a mechanism from one domain to another. Darwin's encounter with Malthus is a canonical example. Darwin was already investigating species transmutation, variation, artificial selection, geographic distribution, and the struggle for existence. Reading Malthus's population argument in 1838 supplied a candidate causal mechanism: reproductive increase under limited resources creates persistent pressure, so inherited variations that improve survival and reproduction tend to be preserved [Malthus, 1798, Darwin, 1958].

Malthusian source structure	Biological transport	Relation proposed to persist
Human population increase	Overproduction of organisms	More individuals arise than available conditions can sustain
Limited means of subsistence	Finite ecological resources and hazards	Population growth encounters checks
Differential exposure to checks	Variation among organisms	Some differences alter persistence and reproduction
Population-level consequence	Change across generations	Preserved heritable variation accumulates in a lineage

Table 22.2: Darwin's Malthusian transfer can be read as a proposed partial functor: a population-pressure mechanism is retyped in a biological domain.

Calling this a *functorial leap* imposes more discipline than calling it an analogy. A proposed partial functor

$$F : \mathcal{C}_{\text{population}} \dashrightarrow \mathcal{C}_{\text{biology}}$$

must state which objects, processes, and relations are transported, which source assumptions have no biological image, and which new consequences follow in the target. The transfer is creative because the target mechanism was not explicit in the source theory or in the biological observations alone. It is scientific because the transported relations can be confronted with variation, inheritance, adaptation, divergence, and extinction [Darwin, 1859].

The historical vocabulary matters. Darwin did not possess the modern concept of a gene, and "survival of the fittest" was introduced by Herbert Spencer and only later adopted by Darwin. The appropriate description is differential preservation and reproductive success of heritable variation. Natural selection was also independently formulated

by Alfred Russel Wallace, leading to their joint communication in 1858 [Darwin and Wallace, 1858].

Closure failure and functorial transfer therefore identify two distinct engines of discovery:

$$\text{missing consequence} \rightsquigarrow \text{completion},$$
$$\text{mechanism in another domain} \rightsquigarrow \text{transport and retyping}.$$

Other engines may require symmetry completion, a new invariant, refinement of an observational cover, or a new primitive that reorganizes the available objects. The larger LINCS problem is to build a typed library of such meta-repairs without pretending that any single obstruction determines the creative proposal.

Design principle

Treat scientific analogy as a proposed partial functor, not as an untyped metaphor. Record the transported structure, the unavoidable mismatches, and the target-domain observations that could reject the transfer.

22.1.5 *Duality as discovery between theories*

The holographic conjecture supplies a third engine of discovery. Maldacena proposed that suitable string or gravitational theories on anti-de Sitter spacetimes are dual to conformal field theories in one fewer noncompact dimension [Maldacena, 1998]. The proposal was surprising because the two descriptions use different primitive vocabularies: one contains gravity and a higher-dimensional bulk, while the other is a nongravitational quantum field theory on its boundary. A quarter-century retrospective emphasizes how little the original proposal resembled an incremental adjustment within either theory [Ananthaswamy, 2023].

This is not naturally represented as adding one missing arrow to a single sketch. Let $\mathbb{S}_{\text{bulk}}$ and $\mathbb{S}_{\text{bdry}}$ present the relevant fragments of the two theories, and let

$$\mathcal{M}_{\text{bulk}} \subseteq \text{Mod}(\mathbb{S}_{\text{bulk}}), \qquad \mathcal{M}_{\text{bdry}} \subseteq \text{Mod}(\mathbb{S}_{\text{bdry}})$$

denote the regimes in which the proposed correspondence is meaningful. The creative conjecture has the form

$$\mathcal{M}_{\text{bulk}} \simeq \mathcal{M}_{\text{bdry}},$$

together with a dictionary transporting states, observables, symmetries, and dynamics. This notation expresses the categorical shape of the claim; it does not assert that every physical formulation of the correspondence has already been realized as an equivalence of ordinary sketch-model categories.

The established evidence concerns anti-de Sitter settings. Our observed universe is closer to a de Sitter cosmology, for which no comparably clear holographic dual is known [Ananthaswamy, 2023].

The clue was structural agreement. The brane field theory and the near-horizon geometry displayed matching symmetry structure, even though their objects and descriptions looked radically different. Subsequent work could then test a growing correspondence dictionary. The duality is especially productive when a strongly coupled problem in one presentation is transported to a tractable regime in the other. Agreement of independently calculated quantities becomes evidence for the proposed equivalence, while a mismatch localizes a gap in the dictionary, an approximation regime, or the conjecture itself.

Discovery object	Characteristic question	Admission evidence
Sketch augmentation $\mathsf{S} \to \mathsf{S}^{+}$	Which generator or axiom is missing from the current theory?	Conservativity, expansion of old models, and novel consequences
Partial functor $\mathcal{C} \dashrightarrow \mathcal{D}$	Does a mechanism from one domain survive retyping in another?	Preserved relations and target-domain tests
Conjectured duality $\mathcal{M}_1 \simeq \mathcal{M}_2$	Do different theories encode the same admissible physics?	A coherent translation dictionary and independently matched observables

Table 22.3: Theory discovery may add structure, transport structure, or identify two presentations through a proposed semantic equivalence.

The LINCS workflow must therefore be able to repair correspondences as well as models and sketches. Declare the two theories and an initial dictionary; differentiate disagreement under perturbations on either side; quotient presentation-dependent descriptions; localize a mismatch to a dictionary entry or validity regime; repair the correspondence; and admit it using observables not employed in proposing the repair. A successful duality can also generate new probes: a construction transparent on one side predicts a previously hidden object or relation on the other.

Design principle

Do not treat a duality as a picturesque analogy. Represent it as a typed, partial correspondence between semantic domains, with an explicit validity regime and a growing set of independently tested translation rules.

22.1.6 AM and degrees of mathematical novelty

The question "Could a machine invent prime numbers?" is a cleaner mathematical analogue of the Einstein test. Merely inventing a new word is not enough. The system must define a concept not explicitly supplied to it, recognize why the concept is interesting, connect it to

existing operations, and use it to formulate consequences that were not built into the prompt.

Lenat's Automated Mathematician, AM, is an important early case [Lenat, 1976]. AM began with 115 incomplete modules for elementary set-theoretic concepts and roughly 250 heuristic rules. It was not given numbers or proof as primitives. Guided by an agenda and concept-interest heuristics, it reconstructed natural numbers, multiplication, factors, and prime numbers. It then formulated familiar claims including unique factorization and Goldbach's conjecture. AM was designed as a concept and conjecture generator rather than a theorem prover, so these claims were not accompanied by proofs.

This is genuine evidence against an overly simple identification of machine novelty with statistical compression. AM enlarged a structured vocabulary, created active concept modules, and explored their relations. Its search was guided by mathematical heuristics and estimates of interestingness, not by a single code-length objective. From the system's perspective, primes were a new concept even though they were a rediscovery from the human perspective.

The example also qualifies the claim. AM's mathematical seed was small, but its inductive bias was not: its supplied concepts, fixed facets, and hundreds of heuristics encoded substantial retrospective knowledge about productive mathematical search. Lenat reported that AM produced no mathematics that was wholly new to humankind on its own, and that its performance deteriorated as it moved farther from its initial conceptual base. Later methodological criticism likewise questioned how much of the reported behavior could be attributed to general principles rather than the program's detailed engineering [Ritchie and Hanna, 1984].

LINCS can separate three levels that are easily conflated. *Lexical novelty* coins a name. *Definitional extension* adds a concept constructible from the current declaration, as primes can be isolated through factorization. *Abductive declaration change* introduces primitives or structural commitments not already generated by the available concept language. AM demonstrated substantial definitional extension; whether its heuristics constituted a mechanism for the third level remains a different question.

AM's architecture also anticipates the LINCS separation between proposal and admission. A new concept module resembles an extension of a declaration; a conjecture declares a candidate composition; examples and counterexamples supply observations; and proof would provide a stronger admission certificate. AM was powerful on the proposal side and deliberately incomplete on the proof side. A LINCS mathematical-discovery system would retain that generative openness while requiring explicit transport, consistency, and proof obligations

before a conjectural extension becomes admitted theory.

Design principle

Do not test concept invention by novel terminology alone. Require a typed definition, nontrivial relations to existing structure, independently checkable consequences, and an explicit distinction between rediscovery, definitional extension, and declaration-level change.

22.2 *What is fixed, and what may be learned?*

A present LINCS instance can be summarized by a declaration

$$\mathfrak{L} = (\mathbb{S}, \mathcal{T}, \chi, \mathcal{U}, \mathcal{R}, \mathcal{A}),$$

where $\mathbb{S}$ is a learning sketch, $\mathcal{T}$ a tangent site, χ a decision or presentation quotient, $\mathcal{U}$ a cover, $\mathcal{R}$ a typed repair language, and $\mathcal{A}$ an admission rule. The candidate model D varies while much of $\mathfrak{L}$ remains fixed.

Future systems may instead move in a category of declarations. A morphism

$$\Phi : \mathfrak{L} \longrightarrow \mathfrak{L}'$$

would have to state how objects, paths, designated cones, probes, null directions, covers, repairs, and admission blocks are transported. It should also say which previously visible failures remain visible after transport.

Definition 22.1 (Conservative declaration change)**.** A change $\Phi : \mathfrak{L} \to \mathfrak{L}'$ is conservative on an audit family $\mathcal{Q}$ when every obstruction witnessed by a registered probe in $\mathcal{Q}$ has a transported witness in $\mathfrak{L}'$. It may expose new obstructions, but it may not erase an old one merely by changing the declaration.

This is only a first safeguard. A useful declaration change must also preserve the intended decision semantics, expose its new blind spots, and be evaluated on held-out audit families. The idea is analogous to conservative extension in logic: a richer language may prove new statements without invalidating the meaning of the old fragment.

Boundary

A learned declaration is not admitted because it makes the current model look more compositional. It is admitted only through evidence independent of the repair that proposed it.

22.2.1 *Theory repair as sketch augmentation*

Chapter 3 interpreted a learning sketch as a categorical theory and a realized system as one of its models. This reveals a second level of the LINCS workflow. Ordinary repair changes $D \in \mathrm{Mod}(\mathbb{S}, \mathcal{C})$. Theory repair proposes a sketch augmentation

$$\iota : \mathbb{S} \longrightarrow \mathbb{S}^{+}$$

and thereby changes the category in which admissible models live. The induced restriction functor

$$\iota^{*} : \mathrm{Mod}(\mathbb{S}^{+}, \mathcal{C}) \longrightarrow \mathrm{Mod}(\mathbb{S}, \mathcal{C})$$

makes the relationship between the two theories inspectable rather than rhetorical.

The fiber of ι^{*} over an old model records its possible expansions. If the fiber is essentially a singleton, the added vocabulary may be only definitional. If it contains several inequivalent expansions, the old theory does not determine the new structure. If it is empty for a previously successful model, the augmentation has rejected that model and must explain why. These cases prevent every successful redescription from being counted as a discovery.

22.2.2 *Appearance, reality, and theoretical invention*

McCarthy argued that learning cannot be reduced to classifying appearances [McCarthy, 2007]. Appearances are partial and dependent on the circumstances of observation, whereas the objects and mechanisms invoked by an explanation are intended to remain coherent across views, times, and actions. In categorical language, an observation functor

$$O : \mathcal{R} \longrightarrow \mathcal{A}$$

maps candidate realities to appearances. An observed $a \in \mathcal{A}$ generally determines a fiber of compatible realities, not a unique inverse image. If the probes available to the learner do not separate that fiber, reality is not identifiable from passive appearance.

The harder case is not selection among already represented latent states but invention of a new theoretical sort. Ancient atomism proposed that visible matter was generated from invisible constituents. Dalton's atomic theory turned that ontological proposal into a constrained explanatory system: elements, atoms, molecules, fixed combinations, and quantitative relations could jointly account for stable regularities in chemical appearance. The theoretical vocabulary was not copied from sense data. It was introduced to make a larger family of observations compose.

This is a declaration-level repair. A persistent obstruction in the old sketch motivates an augmentation

$$\mathbb{S} \longrightarrow \mathbb{S}^{+} = \mathbb{S} + \{\text{new sorts, generators, and laws}\}.$$

The expansion creates a new forward model from theoretical reality to appearance. Its value lies not merely in fitting the observations that motivated it, but in transporting earlier successes and generating consequences that can be tested independently.

Intervention sharpens the test. Distinct candidate realities may agree on passive appearances yet respond differently when experimental conditions are changed. McCarthy's inverse problem and Pearl's intervention problem are therefore successive stages rather than competing diagnoses:

$$\begin{aligned} \text{appearance} &\longrightarrow \text{candidate reality}, \\ \text{candidate reality} &\longrightarrow \text{interventional consequence}, \\ \text{interventional consequence} &\longrightarrow \text{admission or rejection}. \end{aligned}$$

LINCS adds one further possibility: when every candidate inside the current language fails, the repair may change the language itself.

Stage	Model-level workflow	Theory-level workflow
Declare	Fix $\mathbb{S}$ and a candidate model D	Fix the current theory and the language of permitted extensions
Differentiate	Measure how an obstruction moves under model variation	Measure which generators, axioms, or universal properties sustain the failure
Quotient	Remove decision-null or presentation-null model directions	Identify presentation-equivalent extensions and mere definitional changes
Localize	Assign the obstruction to components, contexts, or overlaps	Assign explanatory pressure to missing sorts, arrows, equations, cones, or cocones
Repair	Change the realized maps or representations	Propose $\iota : \mathbb{S} \to \mathbb{S}^{+}$ and compatible expanded models
Admit	Test the repaired model on held-out probes	Test conservativity, old successes, independent evidence, and novel predictions

Table 22.4: The six-stage workflow lifted from model repair to theory augmentation.

At the theory level, differentiation need not mean that sketches form a smooth parameter space. It means constructing a typed sensitivity analysis: which declared relation produces the persistent obstruction, which added generator could mediate it, and which new universal

property would make the candidate explanation testable? Localization therefore returns not only a faulty component but an *explanatory gap* in the current presentation.

The resulting search cannot range indiscriminately over all theories. Accessible-category methods suggest one disciplined possibility: bound the arity and size of candidate generators, construct larger candidates from presentable pieces, and preserve filtered-colimit structure where the semantics requires it [Makkai and Paré, 1989, Adámek and Rosický, 1994]. This does not supply an algorithm for scientific discovery. It supplies a categorical search language in which proposed extensions and their model categories can be compared.

Admission contract

An augmented sketch must do more than absorb the anomaly that proposed it. Admission requires an explicit restriction functor, recovery or principled rejection of successful old models, and consequences tested on evidence not used to construct the augmentation. Novel prediction is stronger evidence than retrospective fit.

In this form, an abductive leap is neither an unexplained jump nor a scalar loss reduction. It is a proposed expansion of the theory's generators and axioms, together with transport obligations and empirical or deductive tests. The examples of quantization, spacetime geometry, natural selection, and latent causal structure can all be read this way, although their appropriate sketch languages will be very different.

22.3 *Adaptive sites and learned covers*

Localization is always relative to a cover. In SID the cover contains foundry contexts and overlaps; in LINCS-Toulmin it contains sources, argument roles, and reasoning steps; in DLINCS it may contain layers, branches, retrieval blocks, or agent subgraphs. A fixed cover can omit exactly the interaction that creates the global obstruction.

An adaptive system should therefore be able to:

1. detect that the current cover is incomplete;
2. propose a refinement or enlargement;
3. transport existing local witnesses to the new cover;
4. test whether newly visible overlaps explain the global failure; and
5. retain an audit trail connecting the two localization regimes.

Let $\mathcal{U} \preceq \mathcal{U}'$ denote a refinement. A local obstruction family $o_{\mathcal{U}}$ should restrict or transport naturally along refinement, while new intersections in $\mathcal{U}'$ can reveal higher-order incompatibilities. The comparison

$$o_{\mathcal{U}} \longrightarrow \mathrm{Res}^{\mathcal{U}'}_{\mathcal{U}}(o_{\mathcal{U}'})$$

tests whether the old diagnosis is recovered inside the richer one.

Design principle

Cover selection should optimize information gained about unresolved obstructions, not simply minimize the measured obstruction on the chosen cover.

The statistical problem resembles active experimental design. A proposed context or overlap has a cost, an expected diagnostic value, and a risk of selection bias. Adaptive covers therefore need uncertainty-aware stopping rules and explicit "coverage not established" outcomes.

22.4 *Higher interaction signatures*

First-order INC asks whether the tangent lift of a declared factorization is inhabited. It does not exhaust infinitesimal structure. Generated directions can interact through brackets, connections, curvature, torsion, or higher jets. The application must declare which of these operations has meaning.

Definition 22.2 (Interaction transport)**.** Let $D : J \to \mathcal{C}$ be an admissible model with factorization problem $\mathrm{Fact}_{\mathsf{S}}(D)$ and interaction signature Ω_D. An interaction transport is a natural, presentation-descending assignment

$$\mathrm{t}_{D,\mathsf{S}} : \mathrm{Obs}(\mathrm{Fact}_{\mathsf{S}}(D)) \rightsquigarrow \mathrm{Obs}_{\Omega,\mathsf{S}}(TD).$$

It maps a base obstruction to a candidate interaction profile. It does not assert that every component identifies a repair or has predictive value after scalarization.

For admissible fields X, Y, bracket INC records relative failure of closure:

$$\mathrm{INC}_{[,]}(D) = \{[X,Y] \mid X, Y \in \mathcal{A}_D,\ [X,Y] \notin \mathcal{A}_D\}.$$

This antisymmetric operation underlies the causal screen used by BRIDGE and SKFM [Mahadevan, 2026b]. Its semantics do not transfer automatically to adapters, policies, or skills.

With a declared connection ∇, the ordered profile

$$J^2_{\nabla,D}(X,Y) = (\nabla_X Y, \nabla_Y X)$$

retains both the antisymmetric component associated with the bracket and the symmetric component associated with joint acceleration. Either component may be irrelevant in a particular sketch. Sparse empirical admission should be allowed to set its coefficient to zero.

Table 22.5: A hierarchy of increasingly structured compositionality tests.

Level	Declared object	Characteristic question	Possible witness
Base	learning diagram D	Do the required finite routes and universal properties hold?	parallel-path or factorization obstruction
Tangent	lifted diagram TD	Does first-order behavior preserve the declaration?	tangent route incompatibility
Interaction	fields with signature Ω_D	Do generated directions close and interact as declared?	bracket, acceleration, torsion, or curvature profile
Higher	prolonged sketch $\mathbb{S}^{(n)}$	Are iterated tangent coherences realized?	flip, jet, Jacobiator, or Bianchi-type obstruction
Homotopical	enriched factorization problem	Can the failure be coherently deformed away?	homotopy or cohomological obstruction class

Weil algebras provide an indexing language for these prolongations [Leung, 2017a,c]. If W is the first-order dual-number object and

$$F : \mathsf{Weil}_1 \longrightarrow \mathrm{End}(\mathcal{C})$$

classifies the tangent structure, then $T = F(W)$. Iterated and mixed Weil probes distinguish repeated directional differentiation from the fiber powers used in the tangent axioms. A mature higher LINCS theory should construct prolonged sketches

$$\mathbb{S}^{(n)} \quad \text{with} \quad \mathrm{INC}^{(n)}(D) = \mathrm{Obs}\big(\mathrm{Fact}_{\mathbb{S}^{(n)}}(T^n D)\big)\,,$$

including canonical flips, vertical lifts, and any declared connection data.

22.5 *Homotopical repair*

Strict factorization can be too rigid for representations, policies, world models, or reasoning graphs whose semantics are invariant under controlled deformation. Strict repair asks for

$$D = \overline{D}q,$$

whereas homotopical repair asks for a coherent deformation

$$D \simeq \overline{D}q.$$

Definition 22.3 (Homotopy INC)**.** In a homotopical, model-categorical, or ∞-categorical enrichment of $\mathcal{C}$, define

$$\mathrm{INC}_{\mathrm{h}}(D) = \mathrm{Obs}(\mathrm{Fact}_{\mathrm{S}}(TD) \text{ up to coherent homotopy}).$$

It vanishes when the tangent factorization problem is inhabited up to the declared notion of coherent deformation.

This definition does not turn every approximate equality into a homotopy. The deformation paths, higher coherence cells, and allowed endpoints belong to the declaration. Otherwise "equivalent up to deformation" becomes an unfalsifiable escape from a failed diagram.

Example 22.4 (Argument repair up to presentation)**.** Two Toulmin realizations may differ in sentence order, lexical choice, or the number of intermediate propositions while preserving claim, warrant, source, qualifier, and rebuttal structure. A homotopical argument sketch could treat these presentation changes as coherent paths. A deformation that silently weakens a qualifier or disconnects a source would leave the admitted component and remain obstructed.

Admission contract

A homotopical repair must exhibit the deformation and verify its coherence, endpoint semantics, and behavior under restriction. A low distance between representations is not a substitute for this certificate.

22.6 *Completion and coalgebraic stabilization*

Repeated diagnosis and repair produce an unfolding

$$D, \quad T_{\mathrm{INC}}D, \quad T_{\mathrm{INC}}^2 D, \quad \dots$$

whose behavior can be studied coalgebraically. Existing final-coalgebra theorems give existence under class-based or accessible set semantics [Aczel and Mendler, 1989, Barr, 1993]. Contractive metric realizations can give convergence under stronger analytic assumptions.

The central comparison problem is to relate a freely generated LINCS completion to the final semantics of the INC unfolding:

$$\mathrm{LINCS}(\mathcal{C}) \longrightarrow \nu T_{\mathrm{INC}}.$$

When is this comparison fully faithful? When is it essentially surjective? Can its hypotheses be checked from a learning sketch rather than supplied externally?

One route is through accessibility. If admissible learning sketches and their models form an accessible category, if T_{INC} preserves the relevant filtered colimits, and if presentable objects generate the required semantics, then free-completion and final-behavior views may meet in one construction [Adámek and Rosický, 1994, Makkai and Paré, 1989].

The analytic frontier is equally important. A contraction factor $\rho < 1$ should ideally be derived from the sketch, tangent structure, observation profile, and admission map. Stochastic variants must accommodate martingale noise, biased tangent estimates, changing covers, and data-dependent contraction. Only then can stabilization become a verifiable certificate rather than an assumed property of the optimizer.

Boundary

A final coalgebra may exist even when no feasible computation reaches a useful repaired system. Conversely, empirical convergence of a scalar loss does not establish stabilization of the typed obstruction unfolding.

22.7 *Reflective repair and reverse transport*

DLINCS requires reverse transport of local obstruction information through shared modules, branches, retrieval systems, and agent graphs. Reverse derivative and reverse tangent categories provide candidate abstractions [Cockett et al., 2020, Cruttwell and Lemay, 2024], but a reverse signal is not yet an admissible forward repair.

Let $\mathcal{B}$ be a diagrammatic-backpropagation procedure that maps a global obstruction to local signals and then to parameter or architectural changes. A *reflective* learning sketch includes $\mathcal{B}$, its restriction maps, and its repair maps as arrows. It can then ask whether

$$\begin{array}{ccc} \text{global obstruction} & \xrightarrow{\mathcal{B}} & \text{global repair} \\ \Big\downarrow{\scriptstyle \mathrm{loc}} & & \Big\downarrow{\scriptstyle \mathrm{loc}} \\ \text{local obstructions} & \xrightarrow[\mathcal{B}_{\mathrm{loc}}]{} & \text{local repairs} \end{array}$$

commutes.

Ordinary diagrammatic backpropagation diagnoses a target computation. Reflective LINCS diagnoses how the learning procedure itself transports and repairs that obstruction. A failure may call for a new path, state variable, module, skip connection, memory interface, or compatibility constraint rather than another parameter step.

Reflection here is structural, not anthropomorphic: does the repair mechanism compose with localization and restriction in the way its own declaration requires?

22.8 Topology of repair spaces

A scalar repair cost orders candidates but hides how qualitatively different repairs are connected. Let $\mathcal{R}_{\mathrm{INC}}(D)$ be a category whose objects are admissible base and tangent repairs and whose morphisms are structure-preserving transformations among them. Its nerve

$$N\mathcal{R}_{\mathrm{INC}}(D)_n = \mathrm{Fun}([n], \mathcal{R}_{\mathrm{INC}}(D))$$

is a simplicial set. Vertices are repairs, edges are transformations, and higher simplices encode coherent chains of transformations.

The geometric realization

$$|N\mathcal{R}_{\mathrm{INC}}(D)|$$

turns the repair category into a classifying space. Connected components can distinguish inequivalent repair regimes; loops can record nontrivial cycles; and higher homotopy or homology can detect coherent obstructions invisible to a scalar ranking. This parallels classifying-space and higher algebraic K-theoretic treatments of causal equivalence [Mahadevan, 2023, 2025b].

The ordinary nerve forgets infinitesimal organization. The Weil nerve suggests retaining a Weil-indexed family of repair categories [MacAdam, 2022]:

$$V \longmapsto \left| N\mathcal{R}^{V}_{\mathrm{INC}}(D) \right|, \qquad V \in \mathsf{Weil}_1.$$

The monoidal unit records base repair topology, the dual-number generator records first-order repair directions, and Weil composites organize higher and mixed prolongations.

A complementary integration problem asks whether an infinitesimal repair algebroid assembles into a groupoid of finite repairs. Failure of the infinitesimal-to-finite comparison would be a new obstruction: a locally coherent repair calculus that cannot be globally executed.

Design principle

Topology should distinguish repair regimes, not decorate a loss landscape. The objects and morphisms of the repair category must retain the typed admission semantics before nerve and realization are applied.

22.9 Statistical and language-valued tangent sites

Most present experiments use deterministic perturbations, controlled synthetic data, or a fixed collection of language probes. Statistical LINCS needs obstruction objects that carry uncertainty before scalarization. Required ingredients include:

- support and overlap conditions for tangent probes;
- simultaneous inference across many paths, intersections, and blocks;
- uncertainty propagation through localization and quotienting;
- sequential validity under adaptive covers and repairs;
- robustness to biased or learned tangent estimators; and
- abstention under weak identification.

Infinitesimal causality makes this discipline especially visible: a smooth response field acquires causal meaning only under explicit intervention, support, and identification assumptions [Mahadevan, 2026f]. Similar care is needed when tangent directions are inferred from model activations rather than supplied by a physical protocol.

Language-valued systems add another layer. Argument roles, propositions, sources, and semantic equivalence classes are estimated by fallible models. A robust language tangent site should separate at least:

1. presentation variation such as paraphrase or reordering;
2. semantic variation that preserves the downstream decision;
3. structural variation that changes a role or restriction map; and
4. evidential variation that changes whether a claim is supported.

Inter-annotator variation, parser uncertainty, model-to-model variation, and context drift should remain visible through the repair decision. Otherwise a language model can certify a gluing repair using the same unsupported interpretation that generated it.

22.10 *Mechanizing the declaration-to-certificate path*

The long-term engineering goal is a compiler whose inputs are learning sketches, tangent sites, quotients, covers, and admission contracts, and whose outputs are executable audits and replayable certificates.

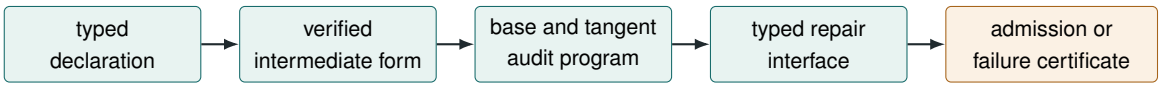


Figure 22.1: A prospective LINCS compiler. Formal verification connects the declaration to generated code; empirical manifests connect the code to the reported evidence.

The compiler should generate:

- base, tangent, and declared higher-order factorization tests;
- localization probes and cover-completeness diagnostics;
- quotient-invariance checks;

- typed repair interfaces with effect annotations;
- blockwise held-out admission tests; and
- provenance-rich certificates for admission, rejection, or abstention.

Chain-of-Evidence demonstrates that provenance-by-construction can already be made operational for autonomous research artifacts [Meng et al., 2026]. A LINCS compiler would extend that base audit to *tangent evidence chains*: certificates that record whether a claim's support survives registered perturbations of data, seeds, evaluators, ablations, and evidence sources, and localize the first structural failure when it does not.

Proof assistants and typed intermediate representations can verify that generated audits match a formal sketch. Experiment manifests serve a different role: they verify that data, probes, seeds, thresholds, and held-out blocks match the empirical claim. Neither layer replaces the other.

Admission contract

A mechanized repair is admitted only when the generated program is faithful to the declaration and the empirical evidence satisfies the registered contract. Successful compilation proves neither condition by itself.

A staged research agenda

Horizon	Theoretical objective	Computational objective	Decisive evidence
Near term	statistical tangent sites and conservative cover refinement	compiler for base/tangent audits and typed certificates	held-out gains with calibrated abstention
Medium term	higher interaction signatures and reverse repair semantics	reflective DLINCS over modular and agentic systems	architectural repairs that outperform parameter-only repair
Medium term	accessible completion and intrinsic stabilization criteria	coalgebraic monitoring of repeated repair	verified convergence of typed obstruction profiles
Long term	homotopical factorization and topology of repair categories	nerve, realization, and infinitesimal-to-finite repair tools	distinct repair regimes and persistent obstruction classes
Long term	functorial transfer across learning domains	portable sketch and admission libraries	guarantees preserved under a declared domain translation
Long term	auditable declaration change and scientific novelty	abductive proposal with active experiment design	resolution of persistent anomalies and successful discriminating predictions

Table 22.6: Frontiers are ordered by the certificates needed to make them scientifically meaningful, not by mathematical novelty alone.

From causal models to actionable learning models

The book can now be read as carrying a central lesson of causal inference into machine learning. Association alone does not determine the effect of an action. A model must expose the relevant mechanisms, specify what an intervention changes, and state what remains invariant. Conventional causal inference applies this discipline to a domain model. LINCS applies it more broadly to the organization of learning itself.

This shift creates a hierarchy of models rather than a competition between causal and noncausal learning. A domain sketch may represent biological, economic, physical, or social mechanisms. A learning sketch represents the system that estimates, composes, compares, and repairs such models. A foundry sketch represents the longer-lived process that gathers evidence, maintains provenance, revises declarations, and decides what enters durable state. Interventions at these levels answer different questions and require different certificates.

The frontier is therefore not merely to attach causal variables to existing predictors. It is to build learning systems whose own components and obligations are explicit enough to support controlled action. Such systems should distinguish:

$$\begin{aligned} \text{change a parameter} &\neq \text{replace a component,} \\ \text{replace a component} &\neq \text{redesign an architecture,} \\ \text{redesign an architecture} &\neq \text{augment a theory.} \end{aligned}$$

The higher the intervention rises, the more demanding the transport and admission problem becomes. A future theory of structural intervention should make these levels compositional, relate their invariance assumptions, and characterize when a repair at one level can be soundly realized at another.

Boundary

LINCS generalizes the model-and-intervention discipline of causal inference; it does not generalize causal semantics by fiat. A learning-system repair is causal only when it is grounded in a mechanism model and a realizable intervention. Otherwise its proper claim is structural, architectural, or theory-level repair.

The continuing LINCS question

The applications in this book show that a common categorical grammar can organize causal discovery, representation learning, adapter composition, skills, reinforcement learning, preferences, relational geometry, predictive descent, and argument repair. The objective is not to force those domains into one universal loss, but to make the translations among their declarations explicit enough to prove what transfers and observe what does not.

The durable LINCS question is therefore:

> Which structural promise failed, how does that failure move, what part of the modeled learning system may be changed while the rest is held invariant, and what independent evidence licenses the repair?

Higher tangents, homotopy, coalgebra, topology, statistical inference, and mechanization each refine a different part of that question. Their value will come from preserving the distinction that motivated LINCS in the first place: an obstruction is a typed failure of composition; a learning signal is an observation of that failure; and a repaired system is one that passes a separate admission decision. The resulting framework is interventionist in method, categorical in its account of structure, and causal only where its semantics and experiments justify that stronger word.

A REPAIR CALCULUS ALSO REVEALS ITS OWN BOUNDARY. LINCS assumes a registered declaration and a registered language of admissible changes. When persistent obstruction cannot be resolved within that language, the problem is no longer only to repair a model of the theory: it is to propose a finite extension of the theory, transport prior knowledge across the extension, and admit the result under independent evidence. The companion volume *Infinitesimal Creativity* begins at this boundary. It treats typed theory extension as a distinct research problem rather than silently promoting an unsuccessful LINCS repair into a creative act.

Further reading

For homotopical extensions, Richter [2020] provides a route from ordinary categories to homotopy theory, while May [1992] develops the simplicial foundations. Accessible and locally presentable categories [Adámek and Rosický, 1994, Makkai and Paré, 1989] are natural settings for studying categories of models, completion, and coalgebraic semantics beyond finite sketches. Universal coalgebra [Rutten, 2000] gives a complementary language for unfolding, behavioral equivalence, and coinductive proof.

Higher tangent structure can be approached through Weil-algebra classifications [Leung, 2017a,c]; MacAdam's functorial treatment connects Weil semantics, Lie theory, nerves, and realization [MacAdam, 2022]. Reverse derivative and reverse tangent categories [Cockett et al., 2020, Cruttwell and Lemay, 2024] frame the reverse-transport problem. On the learning side, categorical deep learning [Gavranović et al., 2024] and copresheaf topological networks [Hajij et al., 2025] suggest how richer declarations might compile into architectures. The open problem for LINCS is to join these theories to uncertainty-aware, mechanized admission without erasing the domain semantics that make an obstruction meaningful.

For the compression view of learning, Rissanen [1978] gives the foundational shortest-description formulation and Grünwald [2007] develops modern MDL in depth. Li and Vitányi [2019] provides the algorithmic-information background, including the invariance and noncomputability qualifications that separate ideal complexity from an operational code. Zahavy [2026] challenges compression as an account of scientific invention by separating inductive compression and deductive verification from abductive changes of axioms. The LINCS question is how such codes should descend through declared quotients and compose across covers, and how candidate declaration changes should be generated and audited, without replacing typed admissibility by a single complexity ranking.

McCarthy [2007] frames a complementary limitation as the gap between appearance and the reality that produces it. His examples connect ordinary object permanence, inverse problems, and the invention of theoretical entities such as atoms. Read together with Pearl's causal critique, this makes explicit why latent explanation and interventional validation are distinct obligations.

Lenat's account of AM [Lenat, 1976] remains a foundational study of heuristic concept formation and conjecture generation. The methodological analysis by Ritchie and Hanna [1984] is a useful counterweight when assessing how much novelty arose from general discovery principles and how much from hand-engineered representation and heuristics. Together they make AM a particularly instructive precursor for LINCS: a system can be strong at proposing extensions while remaining weak at proof, admission, and transfer beyond its initial declaration.

The original papers make the two scientific-discovery loops especially concrete. Planck's radiation-law paper [Planck, 1901] begins from the empirical failure of a previously derived distribution and changes a constitutive assumption in response. Einstein's systematic presentation of general relativity [Einstein, 1916] derives light deflection as a consequence of the new frame, while the eclipse report of Dyson et al. [1920] records the independent observational test. They are useful not as myths of solitary inspiration, but as paired examples of obstruction-driven re-declaration and prediction-driven experiment design.

Malthus's population essay [Malthus, 1798], Darwin's later recollection of reading it [Darwin, 1958], and the mature argument in *On the Origin of Species* [Darwin, 1859] document a different discovery pattern: transport of a mechanism across domains. The joint Darwin–Wallace communication [Darwin and Wallace, 1858] is essential historical context and guards against turning a distributed scientific development into a story of isolated inspiration.

Bibliography

Samson Abramsky and Adam Brandenburger. The sheaf-theoretic structure of non-locality and contextuality. *New Journal of Physics*, 13 (11):113036, 2011.

Peter Aczel and Nax Paul Mendler. A final coalgebra theorem. In *Category Theory and Computer Science*, volume 389 of *Lecture Notes in Computer Science*, pages 357–365. Springer, 1989.

Jiří Adámek and Jiří Rosický. *Locally Presentable and Accessible Categories*, volume 189 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, 1994.

Shun-ichi Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998. DOI: 10.1162/089976698300017746.

Shun-ichi Amari. *Information Geometry and Its Applications*. Springer, 2016. DOI: 10.1007/978-4-431-55978-8.

Anil Ananthaswamy. Is our universe a hologram? Physicists debate famous idea on its 25th anniversary. *Scientific American*, 328(3):58, 2023. DOI: 10.1038/scientificamericano323-58. URL https://doi.org/10.1038/scientificamerican0323-58.

Vladimir Araujo, Marie-Francine Moens, and Tinne Tuytelaars. Learning to route for dynamic adapter composition in continual learning with language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 687–696, 2024. DOI: 10.18653/v1/2024.findings-emnlp.38.

Jose A. Arjona-Medina, Michael Gillhofer, Michael Widrich, Thomas Unterthiner, Johannes Brandstetter, and Sepp Hochreiter. RUDDER: Return decomposition for delayed rewards. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

V. I. Arnold and B. A. Khesin. *Topological Methods in Hydrodynamics*, volume 125 of *Applied Mathematical Sciences*. Springer, New York, 1999. ISBN 978-0-387-94947-5.

Federico Barbero, Cristian Bodnar, Haitz Sáez de Ocáriz Borde, Michael M. Bronstein, Petar Veličković, and Pietro Liò. Sheaf neural networks with connection laplacians. In *Topological, Algebraic and Geometric Learning Workshops*, volume 196 of *Proceedings of Machine Learning Research*, pages 28–36, 2022.

Michael Barr. Terminal coalgebras in well-founded set theory. *Theoretical Computer Science*, 114(2):299–315, 1993. DOI: 10.1016/0304-3975(93)90076-6.

Michael Barr and Charles Wells. On the limitations of sketches. *Canadian Mathematical Bulletin*, 35(3):287–294, 1992. DOI: 10.4153/CMB-1992-040-7. URL https://doi.org/10.4153/CMB-1992-040-7.

Michael Barr and Charles Wells. *Category Theory for Computing Science*. Centre de Recherches Mathématiques, 3 edition, 1999.

Claudio Battiloro, Lucia Testa, Lorenzo Giusti, Stefania Sardellitti, Paolo Di Lorenzo, and Sergio Barbarossa. Generalized simplicial attention neural networks. *IEEE Transactions on Signal and Information Processing over Networks*, 10:1–16, 2024.

Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

Jean Bénabou. Introduction to bicategories. In *Reports of the Midwest Category Seminar I*, volume 47 of *Lecture Notes in Mathematics*, pages 1–77. Springer, Berlin, 1967. DOI: 10.1007/BFb0074299.

Jean-David Benamou and Yann Brenier. A computational fluid mechanics solution to the Monge–Kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, 2000. DOI: 10.1007/s002110050002.

Peter J. Bickel, Chris A. J. Klaassen, Ya'acov Ritov, and Jon A. Wellner. *Efficient and Adaptive Estimation for Semiparametric Models*. Johns Hopkins University Press, 1993.

Stella Biderman, Hailey Schoelkopf, Lintang Sutawika, Baber Abbasi, Jessica Zosa Forde, Leo Gao, Jonathan Tow, Alham Fikri Aji, Pawan Sasanka Ammanamanchi, Sidney Black, et al. Lessons from the trenches on reproducible evaluation of language models, 2026. URL https://arxiv.org/abs/2405.14782.

Cristian Bodnar, Fabrizio Frasca, Nina Otter, Yu Guang Wang, Pietro Liò, Guido Montúfar, and Michael Bronstein. Weisfeiler and lehman go cellular: CW networks. In *Advances in Neural Information Processing Systems*, volume 34, pages 2625–2640, 2021.

Cristian Bodnar, Francesco Di Giovanni, Benjamin P. Chamberlain, Pietro Liò, and Michael M. Bronstein. Neural sheaf diffusion: A topological perspective on heterophily and oversmoothing in GNNs. In *Advances in Neural Information Processing Systems*, volume 35, pages 18527–18541, 2022a.

Cristian Bodnar, Francesco Di Giovanni, Benjamin Paul Chamberlain, Pietro Liò, and Michael M. Bronstein. Neural sheaf diffusion: A topological perspective on heterophily and oversmoothing in GNNs. In *Advances in Neural Information Processing Systems*, volume 35, pages 18527–18541, 2022b.

Jérôme Bolte, Ryan Boustany, Edouard Pauwels, and Béatrice Pesquet-Popescu. On the complexity of nonsmooth automatic differentiation, 2022. URL https://arxiv.org/abs/2206.01730.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models, 2021. URL https://arxiv.org/abs/2108.07258.

Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and Radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51(1):22–45, 2015. DOI: 10.1007/s10851-014-0506-3.

Byron Boots, Sajid M. Siddiqi, and Geoffrey J. Gordon. Closing the learning-planning loop with predictive state representations. *The International Journal of Robotics Research*, 30(7):954–966, 2011.

Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39 (3/4):324–345, 1952.

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.

Matthew Burke and Rory B. B. MacAdam. Involution algebroids: a generalisation of Lie algebroids for tangent categories, 2019. URL https://arxiv.org/abs/1904.06594.

Jérémie Cabessa, Hugo Hernault, and Umer Mushtaq. Argument mining with fine-tuned large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6624–6635. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.coling-main.442/.

Ruichu Cai, Zhiyi Huang, Wei Chen, Zhifeng Hao, and Kun Zhang. Causal discovery with latent confounders based on higher-order cumulants. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 3380–3407. PMLR, 2023. URL https://proceedings.mlr.press/v202/cai23a.html.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, 2018.

N. N. Chentsov. *Statistical Decision Rules and Optimal Inference*, volume 53 of *Translations of Mathematical Monographs*. American Mathematical Society, Providence, R.I., 1982. ISBN 0-8218-4502-0. Translation from the Russian edited by Lev J. Leifman.

Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal*, 21(1):C1–C68, 2018. DOI: 10.1111/ectj.12097.

David Maxwell Chickering. Optimal structure identification with greedy equivalence search. *Journal of Machine Learning Research*, 3: 507–554, 2002.

Kenta Cho and Bart Jacobs. Disintegration and Bayesian inversion via string diagrams. *arXiv preprint arXiv:1709.00322*, 2017. URL https://arxiv.org/abs/1709.00322.

Noam Chomsky. *Aspects of the Theory of Syntax*. MIT Press, Cambridge, MA, 1965. URL https://mitpress.mit.edu/9780262530071/aspects-of-the-theory-of-syntax/.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Kacper Chwialkowski, Heiko Strathmann, and Arthur Gretton. A kernel test of goodness of fit. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2606–2615. PMLR, 2016. URL https://proceedings.mlr.press/v48/chwialkowski16.html.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved

question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018. URL https://arxiv.org/abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. In *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

J. R. B. Cockett and G. S. H. Cruttwell. Differential structure, tangent structure, and SDG. *Applied Categorical Structures*, 22:331–417, 2014a. DOI: 10.1007/s10485-013-9312-0.

J. R. B. Cockett and G. S. H. Cruttwell. Differential structure, tangent structure, and SDG. *Applied Categorical Structures*, 22(2):331–417, 2014b.

J. R. B. Cockett and G. S. H. Cruttwell. The Jacobi identity for tangent categories. *Cahiers de Topologie et Géométrie Différentielle Catégoriques*, 56(4):301–316, 2015. URL https://www.reluctantm.com/gcruttw/publications/jacobiProof.pdf.

J. R. B. Cockett and G. S. H. Cruttwell. Connections in tangent categories. *Theory and Applications of Categories*, 32(26):835–888, 2017.

J. R. B. Cockett and G. S. H. Cruttwell. Differential bundles and fibrations for tangent categories. *Cahiers de Topologie et Géométrie Différentielle Catégoriques*, 59(1):10–92, 2018.

Robin Cockett, Geoffrey Cruttwell, Jonathan Gallagher, Jean-Simon Pacaud Lemay, Benjamin MacAdam, Gordon Plotkin, and Dorette Pronk. Reverse derivative categories. In *Computer Science Logic*, volume 152 of *Leibniz International Proceedings in Informatics*, pages 18:1–18:16, 2020.

Diego Colombo, Marloes H. Maathuis, Markus Kalisch, and Thomas S. Richardson. Learning high-dimensional directed acyclic graphs with latent and selection variables. *The Annals of Statistics*, 40(1):294–321, 2012. DOI: 10.1214/11-AOS940.

Marius Crainic and Rui Loja Fernandes. Integrability of lie brackets. *Annals of Mathematics*, 157(2):575–620, 2003.

G. S. H. Cruttwell, Bruno Gavranovic, Neil Ghani, Paul Wilson, and Fabio Zanasi. Categorical foundations of gradient-based learning. *Programming Languages and Systems*, pages 1–28, 2022. DOI: 10.1007/978-3-030-99336-8_1.

Geoffrey Cruttwell and Jean-Simon Pacaud Lemay. Reverse tangent categories. In *32nd EACSL Annual Conference on Computer Science Logic (CSL 2024)*, volume 288 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 21:1–21:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. DOI: 10.4230/LIPIcs.CSL.2024.21. URL https://arxiv.org/abs/2308.01131.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, volume 26, 2013.

Charles Darwin. *On the Origin of Species by Means of Natural Selection*. John Murray, London, 1859. URL https://darwin-online.org.uk/converted/pdf/1859_Origin_F373.pdf.

Charles Darwin. *The Autobiography of Charles Darwin, 1809–1882*. Collins, London, 1958. URL https://darwin-online.org.uk/content/contentblock?itemID=F1497&viewtype=side&basepage=1&hitpage=73. Edited by Nora Barlow; autobiographical text written 1876–1881.

Charles Darwin and Alfred Russel Wallace. On the tendency of species to form varieties; and on the perpetuation of varieties and species by natural means of selection. *Journal of the Proceedings of the Linnean Society of London. Zoology*, 3(9):45–62, 1858. DOI: 10.1111/j.1096-3642.1858.tb02500.x.

Artur S. d'Avila Garcez, Krysia B. Broda, and Dov M. Gabbay. *Neural-Symbolic Learning Systems: Foundations and Applications*. Springer, London, 2002.

Akash Dhasade, Anne-Marie Kermarrec, Igor Pavlovic, Diana Petrescu, Rafael Pires, Mathis Randl, and Martijn de Vos. Effective LoRA adapter routing using task representations. *arXiv preprint arXiv:2601.21795*, 2026.

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.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using Real NVP. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=HkpbnH9lx.

Arthur Conan Doyle. *The Sign of the Four*. Lippincott's Monthly Magazine, 1890. URL https://www.gutenberg.org/ebooks/2097.

Frank Watson Dyson, Arthur Stanley Eddington, and Charles Davidson. A determination of the deflection of light by the sun's gravitational

field, from observations made at the total eclipse of may 29, 1919. *Philosophical Transactions of the Royal Society of London. Series A*, 220: 291–333, 1920. DOI: 10.1098/rsta.1920.0009.

Stefania Ebli, Michaël Defferrard, and Gard Spreemann. Simplicial neural networks. In *NeurIPS Workshop on Topological Data Analysis and Beyond*, 2020.

Charles Ehresmann. Esquisses et types des structures algébriques. *Buletinul Institutului Politehnic din Iaşi*, 14:1–14, 1968.

Albert Einstein. Die grundlage der allgemeinen relativitätstheorie. *Annalen der Physik*, 354(7):769–822, 1916. DOI: 10.1002/andp.19163540702.

Hayder Elesedy, Pedro M. Esperanca, Silviu Vlad Oprea, and Mete Ozay. LoRA-guard: Parameter-efficient guardrail adaptation for content moderation of large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11746–11765, 2024. DOI: 10.18653/v1/2024.emnlp-main.656.

EleutherAI. Language model evaluation harness. Software, 2026. URL https://github.com/EleutherAI/lm-evaluation-harness.

exo-explore. exo: Run frontier AI locally. Software, 2026. URL https://github.com/exo-explore/exo.

Brendan Fong, David Spivak, and Rémy Tuyéras. Backprop as functor: A compositional perspective on supervised learning. *Proceedings of the 34th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 1–13, 2019.

Tobias Fritz. A synthetic approach to Markov kernels, conditional independence and theorems on sufficient statistics. *Advances in Mathematics*, 370:107239, 2020. DOI: 10.1016/j.aim.2020.107239. URL https://arxiv.org/abs/1908.07021.

Tobias Fritz and Andreas Klingler. The d-Separation criterion in categorical probability. *Journal of Machine Learning Research*, 24(46):1–49, 2023. URL https://jmlr.org/papers/v24/22-0916.html.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 10835–10866, 2023.

Bruno Gavranović, Paul Lessard, Andrew Dudzik, Tamara von Glehn, João G. M. Araújo, and Petar Veličković. Position: Categorical deep learning is an algebraic theory of all architectures, 2024. URL https://arxiv.org/abs/2402.15332.

Steven B. Gillispie and Michael D. Perlman. The size distribution for markov equivalence classes of acyclic digraph models. *Artificial Intelligence*, 141(1–2):137–155, 2002. DOI: 10.1016/S0004-3702(02)00264-3.

Clark Glymour, Kun Zhang, and Peter Spirtes. Review of causal discovery methods based on graphical models. *Frontiers in Genetics*, 10, 2019. ISSN 1664-8021. DOI: 10.3389/fgene.2019.00524. URL https://www.frontiersin.org/journals/genetics/articles/10.3389/fgene.2019.00524.

Christopher Wei Jin Goh, Cristian Bodnar, and Pietro Liò. Simplicial attention networks, 2022.

E Mark Gold. Language identification in the limit. *Information and Control*, 10(5):447–474, 1967. ISSN 0019-9958. DOI: https://doi.org/10.1016/S0019-9958(67)91165-5. URL https://www.sciencedirect.com/science/article/pii/S0019995867911655.

Jackson Gorham and Lester Mackey. Measuring sample quality with kernels. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1292–1301. PMLR, 2017. URL https://proceedings.mlr.press/v70/gorham17a.html.

Peter D. Grünwald. *The Minimum Description Length Principle*. MIT Press, Cambridge, MA, 2007. ISBN 9780262072816.

Siyuan Guo, Viktor Tóth, Bernhard Schölkopf, and Ferenc Huszár. Causal de Finetti: On the identification of invariant causal structure in exchangeable data, 2022. URL https://arxiv.org/abs/2203.15756.

Siyuan Guo, Chi Zhang, Karthika Mohan, Ferenc Huszár, and Bernhard Schölkopf. Do Finetti: On causal effects for exchangeable data, 2024. URL https://arxiv.org/abs/2405.18836.

Ankita Gupta, Ethan Zuckerman, and Brendan O'Connor. Harnessing Toulmin's theory for zero-shot argument explication. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10259–10276, Bangkok, Thailand, 2024. Association for Computational Linguistics. DOI: 10.18653/v1/2024.acl-long.552. URL https://aclanthology.org/2024.acl-long.552/.

Ivan Habernal and Iryna Gurevych. Argumentation mining in user-generated web discourse. *Computational Linguistics*, 43(1):125–179, apr 2017. DOI: 10.1162/COLI_a_00276. URL https://aclanthology.org/J17-1004/.

Elie Habib. Worldmonitor: Real-time global intelligence dashboard. GitHub repository, 2026. URL https://github.com/koala73/worldmonitor. Accessed July 21, 2026.

Mustafa Hajij, Ghada Zamzmi, Theodore Papamarkou, Nina Miolane, Aldo Guzmán-Sáenz, Karthikeyan Natesan Ramamurthy, Tolga Birdal, Tamal K. Dey, Soham Mukherjee, Shreyas N. Samaga, Neal Livesay, Robin Walters, Paul Rosen, and Michael T. Schaub. Topological deep learning: Going beyond graph data, 2022.

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. *arXiv*, 2025. URL https://arxiv.org/abs/2505.21251.

Jakob Hansen and Thomas Gebhart. Sheaf neural networks. In *NeurIPS Workshop on Topological Data Analysis and Beyond*, 2020.

F. Maxwell Harper and Joseph A. Konstan. The MovieLens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems*, 5(4):19:1–19:19, 2015.

Anna Harutyunyan, Will Dabney, Thomas Mesnard, Mohammad Gheshlaghi Azar, Bilal Piot, Nicolas Heess, Hado van Hasselt, Gregory Wayne, Satinder Singh, Doina Precup, and Remi Munos. Hindsight credit assignment. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag, 2nd edition, 2009.

Alain Hauser and Peter Bühlmann. Characterization and greedy learning of interventional Markov equivalence classes of directed acyclic graphs. *Journal of Machine Learning Research*, 13(Aug):2409–2464, 2012.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Ralf Hinze and Dan Marsden. *Introducing String Diagrams: The Art of Category Theory*. Cambridge University Press, 2023. DOI: 10.1017/9781009317825. URL https://doi.org/10.1017/9781009317825.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning*, 2019.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. LoraHub: Efficient cross-task generalization via dynamic LoRA composition. In *Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=lyRpY2bpBh.

Yucong Huang, Xiucheng Li, Kaiqi Zhao, and Jing Li. Transitivity meets cyclicity: Explicit preference decomposition for dynamic large language model alignment, 2026. URL https://arxiv.org/abs/2605.17342.

Aapo Hyvärinen and Stephen M. Smith. Pairwise likelihood ratios for estimation of non-Gaussian structural equation models. *Journal of Machine Learning Research*, 14:111–152, 2013. URL https://www.jmlr.org/papers/v14/hyvarinen13a.html.

Takashi Ikeuchi, Mayumi Ide, Yan Zeng, Takashi N. Maeda, and Shohei Shimizu. Python package for causal discovery based on LiNGAM. *Journal of Machine Learning Research*, 24(14):1–8, 2023. URL https://jmlr.org/papers/v24/21-0321.html.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *International Conference on Learning Representations*, 2023.

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 *Advances in Neural Information Processing Systems*, volume 33, pages 9551–9561, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/6cd9313ed34ef58bad3fdd504355e72c-Abstract.html.

Bart Jacobs, Aleks Kissinger, and Fabio Zanasi. Causal inference by string diagram surgery, 2018. URL https://arxiv.org/abs/1811.08338.

Xiaoye Jiang, Lek-Heng Lim, Yiyu Yao, and Yinyu Ye. Statistical ranking and combinatorial Hodge theory. *Mathematical Programming*, 127(1): 203–244, 2011.

Anatoli Juditsky, Arkadi Nemirovski, and Claire Tauvel. Solving variational inequalities with stochastic mirror-prox algorithm. *Stochastic Systems*, 1(1):17–58, 2011.

Sham M. Kakade. A natural policy gradient. In *Advances in Neural Information Processing Systems 14*. MIT Press, 2001. URL https://proceedings.neurips.cc/paper/2001/hash/4b86abe48d358ecf194c56c69108433e-Abstract.html.

G. M. Kelly. *Basic Concepts of Enriched Category Theory*, volume 64 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, 1982.

Yundong Kim and Heyoung Yang. TRACE: Toulmin-based reasoning assessment through constructive elements for LLM CoT evaluation, 2026. URL https://arxiv.org/abs/2605.29656.

Kimi Team. Kimi K3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026. URL https://arxiv.org/abs/2607.24653.

Anders Kock. *Synthetic Differential Geometry*. Cambridge University Press, 2 edition, 2006. URL https://users-math.au.dk/kock/sdg99.pdf.

Dexter Kozen and Nicholas Ruozzi. Applications of metric coinduction. *Logical Methods in Computer Science*, 5(3:10):1–19, 2009. DOI: 10.2168/LMCS-5(3:10)2009.

Sébastien Lachapelle, Pierre Brouillard, Tristan Deleu, and Simon Lacoste-Julien. Gradient-based neural dag learning. In *International Conference on Learning Representations*, 2020.

LangChain. LangGraph overview. Documentation, 2026. URL https://docs.langchain.com/oss/python/langgraph/overview.

F. William Lawvere. Functorial semantics of algebraic theories. *Proceedings of the National Academy of Sciences*, 50(5):869–872, 1963. DOI: 10.1073/pnas.50.5.869. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC221940/.

F. William Lawvere. Adjointness in foundations. *Dialectica*, 23(3–4): 281–296, 1969.

F. William Lawvere. Toward the description in a smooth topos of the dynamically possible motions and deformations of a continuous body. *Cahiers de Topologie et Géométrie Différentielle Catégoriques*, 21 (4):377–392, 1980. URL https://www.numdam.org/item/CTGDC_1980__21_4_377_0/.

John M. Lee. *Introduction to Smooth Manifolds*, volume 218 of *Graduate Texts in Mathematics*. Springer, 2 edition, 2012. DOI: 10.1007/978-1-4419-9982-5.

Douglas B. Lenat. *AM: An Artificial Intelligence Approach to Discovery in Mathematics as Heuristic Search*. PhD thesis, Stanford University, 1976. URL https://en.wikisource.org/wiki/An_Artificial_Intelligence_Approach_to_Discovery_in_Mathematics_as_Heuristic_Search. Stanford Artificial Intelligence Laboratory Memo AIM-286; Computer Science Report STAN-CS-76-750.

Poon Leung. Classifying tangent structures using Weil algebras. *Theory and Applications of Categories*, 32(9):286–337, 2017a. URL http://www.tac.mta.ca/tac/volumes/32/9/32-09.pdf.

Poon Leung. *Tangent Bundles, Monoidal Theories, and Weil Algebras*. Ph.d. thesis, Macquarie University, 2017b.

Poon Leung. Classifying tangent structures using Weil algebras. *Theory and Applications of Categories*, 32(9):286–337, 2017c. URL http://www.tac.mta.ca/tac/volumes/32/9/32-09.pdf.

Ming Li and Paul Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer, Cham, 4 edition, 2019. DOI: 10.1007/978-3-030-11298-1.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. Diffusion-lm improves controllable text generation. In *Advances in Neural Information Processing Systems*, volume 35, pages 4328–4343, 2022.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Michael L. Littman, Richard S. Sutton, and Satinder Singh. Predictive representations of state. In *Advances in Neural Information Processing Systems*, 2001.

Bo Liu, Ji Liu, Mohammad Ghavamzadeh, Sridhar Mahadevan, and Marek Petrik. Finite-sample analysis of proximal gradient TD algorithms. In *Proceedings of the 31st Conference on Uncertainty in Artificial Intelligence*, pages 504–513, 2015.

Qiang Liu, Jason D. Lee, and Michael I. Jordan. A kernelized Stein discrepancy for goodness-of-fit tests and model evaluation. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 276–284. PMLR, 2016. URL https://proceedings.mlr.press/v48/liub16.html.

Marloes H Maathuis, Markus Kalisch, and Peter Bühlmann. Estimating high-dimensional intervention effects from observational data. *The Annals of Statistics*, 37(6A):3133–3164, 2009.

Saunders Mac Lane. *Categories for the Working Mathematician*. Springer-Verlag, New York, 1971. Graduate Texts in Mathematics, Vol. 5.

Saunders Mac Lane. *Categories for the Working Mathematician*. Springer, second edition, 1998.

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.

Benjamin MacAdam. *The Functorial Semantics of Lie Theory*. PhD thesis, University of Calgary, 2022. URL https://arxiv.org/abs/2301.00305.

Kirill C. H. Mackenzie. *General Theory of Lie Groupoids and Lie Algebroids*, volume 213 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, 2005.

Takashi N. Maeda and Shohei Shimizu. RCD: Repetitive causal discovery of linear non-Gaussian acyclic models with latent confounders. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 735–745. PMLR, 2020. URL https://proceedings.mlr.press/v108/maeda20a.html.

Takashi N. Maeda and Shohei Shimizu. Causal additive models with unobserved variables. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pages 97–106. PMLR, 2021. URL https://proceedings.mlr.press/v161/maeda21a.html.

Hamid Reza Maei. *Gradient Temporal-Difference Learning Algorithms*. PhD thesis, University of Alberta, Edmonton, Alberta, Canada, 2011.

Sridhar Mahadevan. Universal causality. *Entropy*, 25(4):574, 2023. DOI: 10.3390/e25040574. URL https://doi.org/10.3390/e25040574.

Sridhar Mahadevan. Decentralized causal discovery using judo calculus, 2025a. URL https://arxiv.org/abs/2510.23942.

Sridhar Mahadevan. Higher algebraic K-theory of causality. *Entropy*, 27(5):531, 2025b. DOI: 10.3390/e27050531. URL https://doi.org/10.3390/e27050531.

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. ALLORA: A Lie-algebraic LoRA method for composable neural adapters, 2026a. Manuscript in preparation.

Sridhar Mahadevan. Latent confounded causal discovery via Lie bracket geometry, 2026b. URL https://arxiv.org/abs/2606.19610.

Sridhar Mahadevan. Categories for AGI. Book manuscript, 2026c. URL https://people.cs.umass.edu/~mahadeva/papers/catagi.pdf. Revised July 25, 2026.

Sridhar Mahadevan. Causal density functions, 2026d. URL https://arxiv.org/abs/2606.00754.

Sridhar Mahadevan. Gradient infinitesimal reinforcement learning in tangent categories. Manuscript in preparation, 2026e.

Sridhar Mahadevan. Infinitesimal causality. *arXiv preprint arXiv:2606.24621*, 2026f. URL https://arxiv.org/abs/2606.24621.

Sridhar Mahadevan. Kan extension transformers: A categorical unification of attention, diffusion, and predict-detach self-conditioning. *arXiv preprint arXiv:2605.27259*, 2026g. URL https://arxiv.org/abs/2605.27259.

Sridhar Mahadevan. Agentic skill optimization over Lie algebroids. *arXiv preprint arXiv:2607.11493*, 2026h. URL https://arxiv.org/abs/2607.11493.

Sridhar Mahadevan. Learning in infinitesimal non-compositional sketches. arXiv preprint arXiv:2607.15107, 2026i. URL https://arxiv.org/abs/2607.15107.

Sridhar Mahadevan. ODYSSEY: Constructing verifiable, local truth-preserving foundation models, 2026j. Forthcoming arXiv preprint.

Sridhar Mahadevan. PROMETHEUS: Automating deep causal research integrating text, data and models, 2026k. URL https://arxiv.org/abs/2605.12835.

Sridhar Mahadevan. Universal decision learners. *arXiv preprint arXiv:2605.30694*, 2026l. URL https://arxiv.org/abs/2605.30694.

Michael Makkai and Robert Paré. *Accessible Categories: The Foundations of Categorical Model Theory*, volume 104 of *Contemporary Mathematics*. American Mathematical Society, 1989.

Juan M. Maldacena. The large N limit of superconformal field theories and supergravity. *Advances in Theoretical and Mathematical Physics*, 2: 231–252, 1998. DOI: 10.1023/A:1026654312961. URL https://arxiv.org/abs/hep-th/9711200.

Thomas Robert Malthus. *An Essay on the Principle of Population, as It Affects the Future Improvement of Society*. J. Johnson, London, 1798. URL https://www.gutenberg.org/ebooks/4239.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330, 1993.

Dan Marsden. Category theory using string diagrams. *arXiv preprint arXiv:1401.7220*, 2014. URL https://arxiv.org/abs/1401.7220.

James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2408–2417. PMLR, 2015. URL https://proceedings.mlr.press/v37/martens15.html.

Mathematical Association of America. American Invitational Mathematics Examination (AIME) 2024. American Mathematics Competitions problem set, 2024. URL https://maa.org/maa-invitational-competitions/.

J.P. May. *Simplicial Objects in Algebraic Topology*. University of Chicago Press, 1992.

John McCarthy. Challenges to machine learning: Relations between reality and appearance. In *Inductive Logic Programming: 16th International Conference, ILP 2006, Revised Selected Papers*, volume 4455 of *Lecture Notes in Computer Science*, pages 2–9. Springer, 2007. DOI: 10.1007/978-3-540-73847-3_2. URL http://jmc.stanford.edu/articles/appearance/appearance.pdf.

Leland McInnes, John Healy, and James Melville. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Rui Meng, Bhavana Dalvi Mishra, Jiefeng Chen, Chun-Liang Li, Palash Goyal, Mihir Parmar, Yiwen Song, Yale Song, Rajarishi Sinha, Parthasarathy Ranganathan, Burak Gokturk, Jinsung Yoon, and Tomas Pfister. ScientistOne: Towards human-level autonomous research via chain-of-evidence. *arXiv preprint arXiv:2605.26340*, 2026. DOI: 10.48550/arXiv.2605.26340. URL https://arxiv.org/abs/2605.26340.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015. DOI: 10.1038/nature14236.

Ieke Moerdijk and Gonzalo E. Reyes. *Models for Smooth Infinitesimal Analysis*. Springer, New York, 1991. DOI: 10.1007/978-1-4757-4143-8.

Yoshimitsu Morinishi and Shohei Shimizu. Differentiable causal discovery of linear non-Gaussian acyclic models under unmeasured confounding. *Transactions on Machine Learning Research*, 2025. URL https://openreview.net/forum?id=HR7MFlW73I.

Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012. ISBN 978-0262018020.

Ashwin Narayan, Bonnie Berger, and Hyunghoon Cho. Assessing single-cell transcriptomic variability through density-preserving data visualization. *Nature Biotechnology*, 39:765–774, 2021.

Matteo Negro, Andrea Piras, Ragib Ahsan, David Arbour, and Elena Zheleva. Relational causal discovery with latent confounders. In *Proceedings of the Forty-first Conference on Uncertainty in Artificial Intelligence*, volume 286 of *Proceedings of Machine Learning Research*, pages 3123–3154. PMLR, 2025. URL https://proceedings.mlr.press/v286/negro25a.html.

Arkadi Nemirovski. Prox-method with rate of convergence O(1/t) for variational inequalities with Lipschitz continuous monotone operators and smooth convex–concave saddle point problems. *SIAM Journal on Optimization*, 15(1):229–251, 2004.

Allen Newell and Herbert A. Simon. Computer science as empirical inquiry: Symbols and search. *Communications of the ACM*, 19(3): 113–126, 1976. DOI: 10.1145/360018.360022.

Juan Miguel Ogarrio, Peter Spirtes, and Joseph Ramsey. A hybrid causal search algorithm for latent variable models. In *Proceedings of the Eighth International Conference on Probabilistic Graphical Models*, pages 368–379, 2016.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021. URL https://www.jmlr.org/papers/v22/19-1028.html.

Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, USA, 2nd edition, 2009a. ISBN 052189560X.

Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2 edition, 2009b.

Judea Pearl. Theoretical impediments to machine learning with seven sparks from the causal revolution, 2018. URL https://arxiv.org/abs/1801.04016.

Judea Pearl and Dana Mackenzie. *The Book of Why: The New Science of Cause and Effect*. Basic Books, New York, 2018. ISBN 978-0-465-09760-9.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.

Andreas Peldszus and Manfred Stede. An annotated corpus of argumentative microtexts. In *Proceedings of the First European Conference on Argumentation*, 2015. URL https://peldszus.github.io/files/eca2015-preprint.pdf.

Jan Peters and Stefan Schaal. Natural actor-critic. *Neurocomputing*, 71 (7–9):1180–1190, 2008. DOI: 10.1016/j.neucom.2007.11.026.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. AdapterFusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*, pages 487–503, 2021. DOI: 10.18653/v1/2021.eacl-main.39.

Max Planck. Ueber das gesetz der energieverteilung im normalspectrum. *Annalen der Physik*, 309(3):553–563, 1901. DOI: 10.1002/andp.19013090310.

Clifton Poth, Hannah Sterz, Indraneil Paul, Sukannya Purkayastha, Leon Engländer, Timo Imhof, Ivan Vulić, Sebastian Ruder, Iryna Gurevych, and Jonas Pfeiffer. Adapters: A unified library for parameter-efficient and modular transfer learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 149–160, 2023. DOI: 10.18653/v1/2023.emnlp-demo.13.

Akshara Prabhakar, Yuanzhi Li, Karthik Narasimhan, Sham Kakade, Eran Malach, and Samy Jelassi. LoRA soups: Merging LoRAs for practical skill composition tasks. In *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*, pages 644–655, 2025.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741, 2023.

Balaraman Ravindran. *An Algebraic Approach to Abstraction in Reinforcement Learning*. PhD thesis, University of Massachusetts Amherst, 2004.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof Q&A benchmark. *arXiv preprint arXiv:2311.12022*, 2023. URL https://arxiv.org/abs/2311.12022.

Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1530–1538. PMLR, 2015. URL https://proceedings.mlr.press/v37/rezende15.html.

B. Richter. *From Categories to Homotopy Theory*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2020. ISBN 9781108479622. URL https://books.google.com/books?id=pnzUDwAAQBAJ.

Emily Riehl. *Category Theory in Context*. Dover Publications, 2017.

Jorma Rissanen. Modeling by shortest data description. *Automatica*, 14 (5):465–471, 1978. DOI: 10.1016/0005-1098(78)90005-5.

Graeme D. Ritchie and F. Keith Hanna. AM: A case study in AI methodology. *Artificial Intelligence*, 23(3):249–268, 1984. DOI: 10.1016/0004-3702(84)90015-8.

James M. Robins, Andrea Rotnitzky, and Lue Ping Zhao. Estimation of regression coefficients when some regressors are not always observed. *Journal of the American Statistical Association*, 89(427):846–866, 1994. DOI: 10.1080/01621459.1994.10476818.

R. Tyrrell Rockafellar. Generalized directional derivatives and subgradients of nonconvex functions. *Canadian Journal of Mathematics*, 32(2): 257–280, 1980. DOI: 10.4153/CJM-1980-020-7.

Jiří Rosický. Abstract tangent functors. *Diagrammes*, 12:1–11, 1984.

Donald B. Rubin. Causal inference using potential outcomes: Design, modeling, decisions. *Journal of the American Statistical Association*, 100 (469):322–331, 2005. DOI: 10.1198/016214504000001880.

Jan J. M. M. Rutten. Universal coalgebra: a theory of systems. *Theoretical Computer Science*, 249(1):3–80, 2000.

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.

Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *The Semantic Web*, pages 593–607. Springer, 2018.

Dominik Schmid and Allan Sly. On the number and size of markov equivalence classes of random directed acyclic graphs. arXiv:2209.04395v2 [math.PR], 2024. URL https://arxiv.org/abs/2209.04395. Submitted 2022; revised 2024. DOI: 10.48550/arXiv.2209.04395.

John Schulman, Sergey Levine, Pieter Abbeel, Michael I. Jordan, and Philipp Moritz. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pages 1889–1897, 2015.

A. Sepúlveda-Jiménez. Synthetic machine learning. ResearchGate preprint, July 2025. URL https://doi.org/10.13140/RG.2.2.26283.96802.

Abhishek Sharma, Leo Benac, Sonali Parbhoo, and Finale Doshi-Velez. Decision-point guided safe policy improvement. In *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pages 2935–2943. PMLR, 2025. URL https://proceedings.mlr.press/v258/sharma25a.html.

Shohei Shimizu, Patrik O. Hoyer, Aapo Hyvärinen, and Antti Kerminen. A linear non-Gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7:2003–2030, 2006. URL https://www.jmlr.org/papers/v7/shimizu06a.html.

Shohei Shimizu, Takanori Inazumi, Yasuhiro Sogawa, Aapo Hyvärinen, Yoshinobu Kawahara, Takashi Washio, Patrik O. Hoyer, and Kenneth Bollen. DirectLiNGAM: A direct method for learning a linear non-Gaussian structural equation model. *Journal of Machine Learning Research*, 12:1225–1248, 2011. URL https://www.jmlr.org/papers/v12/shimizu11a.html.

Satinder Singh, Michael R. James, and Matthew R. Rudary. Predictive state representations: A new theory for modeling dynamical systems. *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, 2004a.

Satinder Singh, Michael R. James, and Matthew R. Rudary. Predictive state representations: A new theory for modeling dynamical systems. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, pages 512–519, 2004b.

Paul Smolensky. On the proper treatment of connectionism. *Behavioral and Brain Sciences*, 11(1):1–23, 1988. DOI: 10.1017/S0140525X00052432.

Vishal Soni and Satinder Singh. Abstraction in predictive state representations. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence*, pages 639–644, 2007.

Elizabeth S. Spelke. Core knowledge. *American Psychologist*, 55(11): 1233–1243, 2000. DOI: 10.1037/0003-066X.55.11.1233. URL https://doi.org/10.1037/0003-066X.55.11.1233.

Elizabeth S. Spelke and Katherine D. Kinzler. Core knowledge. *Developmental Science*, 10(1):89–96, 2007. DOI: 10.1111/j.1467-7687.2007.00569.x. URL https://doi.org/10.1111/j.1467-7687.2007.00569.x.

Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. MIT Press, 2 edition, 2000.

David I. Spivak. Functorial data migration. *Information and Computation*, 217:31–51, 2012. DOI: 10.1016/j.ic.2012.05.001.

David I. Spivak. Database queries and constraints via lifting problems. *Mathematical Structures in Computer Science*, 24(6):e240602, 2014. DOI: 10.1017/S0960129513000479. URL https://doi.org/10.1017/S0960129513000479.

Ross Street. Categorical structures. In *Handbook of Algebra*, volume 1, pages 529–577. Elsevier, 1996. DOI: 10.1016/S1570-7954(96)80019-2.

Ross Street. Frobenius monads and pseudomonoids. *Journal of Mathematical Physics*, 45(10):3930–3948, 2004. DOI: 10.1063/1.1788852.

Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. *Density Ratio Estimation in Machine Learning*. Cambridge University Press, 2012. DOI: 10.1017/CBO9781139035613.

Richard S. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3:9–44, 1988. DOI: 10.1007/BF00115009.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. Adaptive Computation and Machine Learning Series. The MIT Press, second edition, 2018. ISBN 9780262039246. URL http://incompleteideas.net/book/the-book.html.

Richard S. Sutton, Csaba Szepesvári, and Hamid Reza Maei. A convergent O(n) temporal-difference algorithm for off-policy learning with linear function approximation. *Advances in Neural Information Processing Systems*, 21, 2008.

Gokul Swamy, Christoph Dann, Rahul Kidambi, Steven Wu, and Alekh Agarwal. A minimaximalist approach to reinforcement learning from human feedback. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pages 47345–47377, 2024.

Dennis Tang, Prateek Yadav, Yi-Lin Sung, Jaehong Yoon, and Mohit Bansal. LoRA merging with SVD: Understanding interference and preserving performance. In *ICML Workshop on Resource-Efficient Foundation Models*, 2025.

Tatsuya Tashiro, Shohei Shimizu, Aapo Hyvärinen, and Takashi Washio. ParceLiNGAM: A causal ordering method robust against latent confounders. *Neural Computation*, 26(1):57–83, 2014. DOI: 10.1162/NECO_a_00533.

Philip S. Thomas, William Dabney, Stephen Giguere, and Sridhar Mahadevan. Projected natural actor-critic. In Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2337–2345, 2013. URL https://proceedings.neurips.cc/paper/2013/hash/dd77279f7d325eec933f05b1672f6a1f-Abstract.html.

Stephen E. Toulmin. *The Uses of Argument*. Cambridge University Press, 1958.

Daniele Tramontano, Yaroslav Kivva, Saber Salehkaleybar, Mathias Drton, and Negar Kiyavash. Causal effect identification in lvLiNGAM from higher-order cumulants. *arXiv preprint arXiv:2506.05202*, 2025. URL https://arxiv.org/abs/2506.05202.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting, 2023. URL https://arxiv.org/abs/2305.04388.

Ruben van Belle. *Kan Extensions in Probability Theory*. PhD thesis, University of Edinburgh, 2024.

Alexander Van-Brunt and Matt Visser. Special cases of the baker–campbell–hausdorff formula. *Journal of Mathematical Physics*, 57(2), 2016. DOI: 10.1063/1.4940799.

Mark J. van der Laan and Sherri Rose. *Targeted Learning: Causal Inference for Observational and Experimental Data*. Springer, 2011. DOI: 10.1007/978-1-4419-9782-1.

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.

Cédric Villani. *Optimal Transport: Old and New*, volume 338 of *Grundlehren der mathematischen Wissenschaften*. Springer, Berlin, 2009. DOI: 10.1007/978-3-540-71050-9.

Eric Wallace, Nicholas Tomlin, Albert Xu, Kevin Yang, Eshaan Pathak, Matthew Ginsberg, and Dan Klein. Automated crossword solving. *arXiv preprint arXiv:2205.09665*, 2022.

Yingfan Wang, Haiyang Huang, Cynthia Rudin, and Yaron Shaposhnik. Understanding how dimension reduction tools work: An empirical approach to deciphering t-SNE, UMAP, TriMap, and PaCMAP for data visualization. *Journal of Machine Learning Research*, 22(201):1–73, 2021.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi

Fan, Xiang Yue, and Wenhu Chen. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2406.01574.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2022. URL https://arxiv.org/abs/2201.11903.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems*, 2023.

Yifan Yang, Ziyang Gong, Weiquan Huang, Qihao Yang, Ziwei Zhou, Zisu Huang, Yan Li, Xuemei Gao, Qi Dai, Bei Liu, Kai Qiu, Yuqing Yang, Dongdong Chen, Xue Yang, and Chong Luo. SkillOpt: Executive strategy for self-evolving agent skills, 2026. URL https://arxiv.org/abs/2605.23904.

Deniz Yeral. Frobenius algebras, factorization homology and the Reshetikhin–Turaev invariants, 2025. URL https://arxiv.org/abs/2508.16351.

Olga Zaghen, Antonio Longa, Steve Azzolin, Lev Telyatnikov, Andrea Passerini, and Pietro Liò. Sheaf diffusion goes nonlinear: Enhancing GNNs with adaptive sheaf laplacians. In *Proceedings of the Geometry-grounded Representation Learning and Generative Modeling Workshop*, volume 251 of *Proceedings of Machine Learning Research*, pages 264–276. PMLR, 2024. URL https://proceedings.mlr.press/v251/zaghen24a.html.

Tom Zahavy. Position: LLMs can't jump. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=klU4737opt. Position paper.

Haobo Zhang and Jiayu Zhou. Unraveling LoRA interference: Orthogonal subspaces for robust model merging. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pages 26459–26472. Association for Computational Linguistics, 2025. DOI: 10.18653/v1/2025.acl-long.1284. URL https://aclanthology.org/2025.acl-long.1284/.

Haopeng Zhang, Xiao Liu, and Jiawei Zhang. HEGEL: Hypergraph transformer for long document summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10167–10176, 2022.

Juzheng Zhang, Jiacheng You, Ashwinee Panda, and Tom Goldstein. LoRI: Reducing cross-task interference in multi-task low-rank adaptation. In *Conference on Language Modeling*, 2025a. URL https://openreview.net/forum?id=b8cW86QcOD.

Yifan Zhang, Ge Zhang, Yue Wu, Kangping Xu, and Quanquan Gu. Beyond Bradley–Terry models: A general preference model for language model alignment. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pages 76939–76965, 2025b.

Ziyu Zhao, Leilei Gan, Guoyin Wang, Wangchunshu Zhou, Hongxia Yang, Kun Kuang, and Fei Wu. LoraRetriever: Input-aware LoRA retrieval and composition for mixed tasks in the wild. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 4447–4462, 2024. DOI: 10.18653/v1/2024.findings-acl.263.

Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric Xing. Dags with no tears: Continuous optimization for structure learning. *Advances in Neural Information Processing Systems*, 2018.

Yujia Zheng, Biwei Huang, Wei Chen, Joseph Ramsey, Mingming Gong, Ruichu Cai, Shohei Shimizu, Peter Spirtes, and Kun Zhang. Causal-learn: Causal discovery in python. *Journal of Machine Learning Research*, 25(60):1–8, 2024. URL http://jmlr.org/papers/v25/23-0237.html.

Glossary of Notation

The same letter may denote different objects in different application chapters. This glossary records the stable conventions of the LINCS framework. Types, declarations, and local definitions in the chapters take precedence whenever a symbol is specialized.

Categorical declarations

Notation	Meaning	Role in LINCS
$\mathcal{C}, \mathcal{D}$	categories	Typed domains of objects and composable maps.
$f : X \to Y$	morphism	A legal computation, translation, observation, or intervention channel.
$g \circ f$	composite	The structural operation whose declared coherence is tested.
$D : J \to \mathcal{C}$	diagram	A typed family of objects and maps with shape J.
S	sketch or structural declaration	Generators, equations, cones, cocones, and other obligations imposed before choosing a loss.
$\mathrm{Mod}(\mathsf{S}, \mathcal{C})$	models of S in $\mathcal{C}$	Realizations satisfying the declared sketch structure.
$\mathrm{Path}(\mathsf{S})$	path category of a sketch	Formal composites generated by the declared arrows.
$p, q : X \rightrightarrows Y$	parallel paths	Two computations declared or tested to agree.
Fact	factorization comparison	The typed test used to compare a direct arrow with a declared composite.
$F : \mathcal{C} \to \mathcal{D}$	functor	A compositional translation between structural domains.
$\eta : F \Rightarrow G$	natural transformation	A coherent change of functorial realization.
$(\mathcal{V}, \otimes, I)$	enriching symmetric monoidal category	Supplies the type of hom-objects and the tensor used to compose them.
$\mathcal{C}(X, Y) \in \mathcal{V}$	enriched hom-object	Records structured maps from X to Y, such as an order, distance, or vector space of maps.
$\mathsf{S}_{\mathcal{V}}$	$\mathcal{V}$-enriched sketch	A compositional declaration using enriched equations and designated weighted limits or colimits.
Ω	subobject classifier	Internal object of truth values; its Heyting-algebra structure need not be Boolean.
$\chi_A : X \to \Omega$	characteristic map of $A \hookrightarrow X$	Classifies a predicate or context-dependent subobject without assuming that it has a decidable complement.

Graphical convention. In the 2-categorical figures, a region denotes a category, a wire separating two regions denotes a functor from the region on its left to the region on its right, and a box denotes a natural transformation read from bottom to top. Horizontal pasting is whiskering; vertical stacking is composition of natural transformations. Teal wires mark tangent structure and amber boxes mark comparison or

diagnostic cells. Color is explanatory and carries no additional type information.

Obstruction, observation, and scalarization

Notation	Meaning	Role in LINCS
$\mathcal{O}(D)$	typed obstruction of realization D	Records failure of a declared factorization or universal property.
$\mathcal{O}_{p,q}(D)$	obstruction associated with parallel paths p, q	Localizes the failed promise to a particular comparison.
Obs	observer	Maps a typed obstruction into evidence available to a learner or controller.
$\phi(\mathcal{O})$	observation or feature of an obstruction	May retain vector-, relation-, distribution-, or language-valued information.
$\ell(\phi(\mathcal{O}))$	scalarization	A task-dependent ranking or measurement applied after structural typing.
INC	infinitesimal non-compositionality	Tangent response of a compositional obstruction under an admitted probe.
$\mathrm{INC}_{p,q}$	path-specific infinitesimal obstruction	Directional failure associated with the pair p, q.
λ	scalar weighting parameter	Used only when an admitted realization combines structural observations with scalar objectives.
ker(Obs)	observational null directions	Variations that the declared observer cannot distinguish.

Tangents, probes, and infinitesimal structure

Notation	Meaning	Role in LINCS
$(\mathcal{C}, \mathbb{T})$	tangent category with chosen tangent structure $\mathbb{T} = (T, p, 0, +, \ell, c)$	Encodes admitted infinitesimal variation and its coherent transport.
$T : \mathcal{C} \to \mathcal{C}$, TX	tangent functor and tangent object	Maps an object to its object of first-order directions.
$T_n X$	n-fold fiber power $TX \times_X \cdots \times_X TX$	Collects tangent vectors sharing one base point; $T_2 X$ is the domain of addition.
$T^n X$	n-fold iterate of T	Represents iterated tangent levels; it is not the fiber power $T_n X$.
$p, 0, +, \ell, c$	projection, zero, fiberwise addition, vertical lift, and canonical flip	Structural maps constrained by the tangent-category axioms.
$Tf : TX \to TY$	tangent lift of f	Transports infinitesimal variation through a declared map.
D_ε	infinitesimally perturbed realization	A probe-indexed family with base point D_0.
$\varepsilon^2 = 0$	first-order nilpotent relation	Synthetic differential-geometric encoding of first-order variation.
W	Weil algebra or probe object	Specifies the admitted order and shape of infinitesimal variation.
δ	tangent direction or probe	A local variation whose semantics must be declared.
$[X, Y]$	Lie bracket of vector fields or infinitesimal actions	Detects order-sensitive interaction and failure of involutive closure.
$\mathbf{Db}_{\mathcal{C}}(\mathcal{S}) = [\mathcal{S}, \mathcal{C}]$	category of $\mathcal{C}$-valued database instances	Carries pointwise tangent structure when $\mathcal{C}$ is a tangent category.
$T_{\mathcal{S}} I = T \circ I$	pointwise tangent lift of a database instance	Differentiates instance values while leaving the schema fixed.
$\tau\mathcal{S}$	syntactic tangent expansion of a database schema	Declares tangent objects, fields, brackets, and their constraints explicitly.
$\mathcal{O}_{\mathrm{br}}$	bracket factorization obstruction	Records failure of the visible database distribution to close.
$\rho : A \to TM$	anchor of a Lie algebroid	Maps abstract skills or generators to vector fields on a base manifold.
$\mathcal{N}$	null or gauge subspace	Directions quotiented before diagnosis when they are irrelevant to the declared decision.
$g^\flat : TM \to T^*M$	declared tangent metric as a bundle morphism	Converts tangent vectors to covectors; its inverse $g^\sharp$, when admitted, selects intrinsic repair directions.
h	metric on obstruction fibers	Measures the residual of an inexact typed repair without defining the obstruction itself.
$\xi^\star_\theta$	Natural LINCS tangent repair	Least intrinsic model variation that cancels, or approximately contracts, the linearized typed obstruction.
F_θ	Fisher information metric	Information-geometric choice of g_θ for a declared stochastic model and sampling law.

Repair and admission

Notation	Meaning	Role in LINCS
$\mathcal{U} = \{U_i \to U\}$	cover	Contexts over which an obstruction is localized or evidence is assembled.
$D/\sim$	quotient realization	Removes declared gauge, presentation, or decision-irrelevant variation.
$\mathrm{Loc}(\mathcal{O})$	localization of an obstruction	Assigns surviving failure to modules, overlaps, agents, or contexts.
$\mathcal{R}$	repair language	Registered class of typed changes that may be proposed.
$r : D \rightsquigarrow D'$	repair proposal	A controlled structural change, not yet an admitted update.
$\mathcal{A}$	admission rule or contract	Independent criteria for accepting, rejecting, quarantining, or deferring a repair.
$\mathrm{Cert}(r)$	repair certificate	Provenance, invariants, uncertainty, tests, and evidence associated with a proposal.
$\mathfrak{C}_k$	CoLT reasoning state	Packages the maintained sketch and realization with its probes, quotient, cover, repair language, admission rule, and evidence registry.
τ_k	CoLT transition record	Records a probe, typed obstruction, localization, repair proposal, certificate, and admission decision.
SCoLT	Sheaf-theoretic Chain of LINCS Thought	Specializes CoLT to the Toulmin argument sheaf supplied by SCoTT.
ESS	effective sample size	Statistical admission diagnostic in weighted or off-policy settings.
$\mathrm{Lan}_J F$	left Kan extension	Universal extension by colimit-like aggregation.
$\mathrm{Ran}_J F$	right Kan extension	Universal extension by limit-like consistency.

Local-to-global, causal, and decision notation

Notation	Meaning	Role in LINCS
$\mathcal{F}(U)$	sections over context U	Local predictions, arguments, evidence, or model fragments.
ρ_V^U	restriction map	Transports a section on U to $V \subseteq U$.
$\check{C}^\bullet(\mathcal{U}, \mathcal{F})$	Cech complex of a cover	Records overlap compatibility and higher gluing information.
$\delta_{\check{C}}$	Cech coboundary	Measures failure of local data to agree on overlaps.
colim, lim	colimit and limit	Universal aggregation and compatibility constructions.
$P(Y \mid X = x)$	observational conditional	Evidence obtained by conditioning, without an intervention claim.
$P(Y \mid \mathrm{do}(X = x))$	interventional distribution	A measure defined under a declared intervention mechanism.
$D_{\mathrm{KL}}(P\|Q)$	Kullback–Leibler divergence	Compares identified measures but does not certify causal identification.
V^π, Q^π	value functions	Decision quantities under policy π.
T^π, T^*	Bellman operators	Policy and optimality factorizations used in decision and RL chapters.
$\mathrm{Ran}_J(J^* \mathrm{Lan}_J F)$	universal decision composite	Extension followed by consistency relative to the information made visible by J.

Boundary: Typed notation

A symbol does not acquire causal, decision, or repair semantics from its shape alone. Its type, declaration, observer, and admission contract determine what the notation licenses in a particular application.

Subject Index