Functors, Not Fragments: A Non-Reductionist Methods Program for Science, Mind, and Meaning

Douglas C. Youvan

doug@youvan.com

www.youvan.ai

October 5, 2025

This paper proposes a constructive, falsifiable alternative to reductionism. Instead of dissolving higher strata (life, mind, person, covenant) into lower strata (particles and fields), we model strata as real domains with their own norms and bridge them by structure-preserving maps. The core claim is methodological: explanations should commute. When we pass information across scales or vocabularies, the story must remain consistent under intervention. We formalize this with a role calculus (constraints, dynamics, sources), sheaf semantics for multi-scale coherence, and adjoint coarse-grain/lift operators that capture downward guidance without violating physics. To keep the program empirical, we introduce two artifacts: Logos-Interpreter, a benchmark that scores diagram-commutation and out-of-distribution generalization, and Teleology-Finder, a tool that infers end-congruent constraints and purpose-consistent policies from learned models. We prove conservativity where possible (higher-level additions do not contradict established lower-level theorems) and propose predictive bets that would count against our view if they fail. The result is neither mysticism nor mechanistic erasure: it is a disciplined way to let purposes, promises, and persons enter scientific explanation without collapsing them into nothing-but. If successful, this program yields explanations that generalize better, align more safely, and speak truth across strata.

Keywords: non-reductionism, functors, commutation, sheaf semantics, adjoint functors, teleology, constraints dynamics sources, conservativity, interpretability, alignment, category theory, HoTT, multi-scale modeling, downward causation, benchmark design.

A collaboration with GPT-5. CC4.0. 58 pages.

Outline

1. Motivation: Beyond “Nothing-But”
   1.1 The limits of reduction as dissolution
   1.2 Layered realism: physics -> life -> mind -> person -> covenant
   1.3 Explanations that commute vs explanations that fit

2. Principles Without Reduction
   2.1 Constraint-first causation: formal and final causes alongside efficient causes
   2.2 Roles over parts: constraints, dynamics, sources (CDS)
   2.3 Conservativity: extending without contradiction

3. Mathematics of Bridging, Not Flattening
   3.1 Explanations as functors; naturality as commutation
   3.2 Adjoint pairs for coarse-grain and lift (upward/downward guidance)
   3.3 Sheaf semantics for multi-scale coherence and gluing tests
   3.4 Copy-safe math conventions and notation (ASCII-safe)

4. The Role Calculus (CDS)
   4.1 Definitions: constraint sets, dynamical laws, source terms
   4.2 Intervention semantics and commutation error
   4.3 Identifiability: role stability across model classes

5. Logos-Interpreter (Benchmark)
   5.1 Tasks: diagram-commuting explanations under interventions
   5.2 Metrics: commutation error, OOD generalization, auditability
   5.3 Baselines and ablations
   5.4 Failure modes and diagnostic reports

6. Teleology-Finder (Tool)
   6.1 Inferring ends: extracting purpose-consistent constraints
   6.2 From learned models to minimal end-congruent programs
   6.3 Case studies: biology, cognition, cooperative agents
   6.4 Guarantees and limits

7. Downward Guidance Without Physics Violations
   7.1 Adjoint laws: coarse-grain $\dashv$ lift as formalizing guidance
   7.2 Worked examples: anesthesia, promise-keeping in agents, ecosystem control
   7.3 Conservativity proofs and counterexamples

8. Empirical Program and Predictive Bets
   8.1 Where non-reductive models should win (and why)

1. Motivation: Beyond “Nothing-But”

1.1 The limits of reduction as dissolution

Reduction becomes dissolution when the explanans erases the explanandum. Three recurrent failure modes:

- **Category mistake:** Explaining promise-keeping as “nothing-but dopamine” swaps a norm-governed act (truth under obligation) for a mechanism that sometimes implements it. Mechanism can enable or disable fidelity; it does not equal fidelity.
- **Predictive brittleness:** Purely micro-level fits often fail under intervention or domain shift. A model that nails lab data collapses in the field because it never represented the higher-level invariants (roles, constraints, ends) that actually travel.
- **Loss of evaluative content:** Explanations that delete thick predicates (true, just, faithful, responsible) cannot guide action in domains where those predicates are operative. Science then under-explains precisely where society needs guidance.

We retain everything reduction promises—parsimony, testability, mechanistic detail—without the “nothing-but.” Our stance: keep lower-level accounts, add the higher-level invariants they instantiate, and require mappings between grains to **preserve structure** rather than dissolve it.

Operational test for dissolution:

- If replacing a higher-level predicate H with a lower-level surrogate L makes it impossible to state when an intervention succeeded (e.g., “kept the promise” vs “raised reward”), the account has dissolved H.
- Remedy: introduce role-preserving maps so that changes measured below still decide H above.

1.2 Layered realism: physics -> life -> mind -> person -> covenant

We assume strata that are real and normed:

- **Physics:** Efficient causes, symmetries, conservation; lawful dynamics over states.
- **Life:** Teleonomy, function, regulation; constraint architectures that maintain far-from-equilibrium order.
- **Mind:** Representation, attention, inference; correctness and mistake become meaningful predicates.
- **Person:** Address, promise, responsibility; answerability to reasons, not just causes.

- **Covenant:** Public commitments that bind persons and institutions over time; justice, fidelity, mercy as governing norms.

Each layer introduces new invariants that are not eliminable but are **implementable** by the layer below. The relation is not "mind floats free of matter," but "mind's norms are realized by material dynamics in a way that preserves their distinct evaluative content."

Method rule:

- Treat each layer as a category C_k with objects (systems, acts, institutions) and morphisms (transformations, histories). Lower layers provide realizers; higher layers provide admissibility and evaluation.

Working examples:

- **Anesthesia:** Physics-level perturbation (ligands, ion channels) disables mind-level invariants (reportable contents), yet personal identity and covenants persist across the gap. Layered realism lets us say why all three statements can be true without contradiction.
- **Alignment:** A reward-maximizing policy can implement cooperation transiently; a promise-keeping policy carries a different invariant (keep commitments under pressure). The higher-layer predicate allows audits and interventions that pure reward cannot distinguish.

1.3 Explanations that commute vs explanations that fit

Explanations that fit minimize error on observed data at some grain. **Explanations that commute** preserve the diagram of relations between grains under interventions. Formally, let:

- E_k be an explanans at layer k (e.g., mechanistic model at physics, normative model at person).
- T_{k->k+1} be an upward map (coarse-grain, abstraction).
- S_{k+1->k} be a downward map (lift, guidance/implementation).
- I_k be an intervention at layer k (e.g., pharmacological dose; contractual clause).

A commuting explanation satisfies, for relevant families of interventions,

T_{k->k+1} \circ E_k \circ I_k ~= E_{k+1} \circ T_{k->k+1} \circ I_k

S_{k+1->k} \circ E_{k+1} \circ I_{k+1} ~= E_k \circ S_{k+1->k} \circ I_{k+1}

where “~=“ denotes agreement within pre-registered tolerance. Intuition: whether we explain before or after moving across layers, the predicted effect is the same. The story **travels**.

Practical payoffs:

- **OOD generalization:** Models that commute across strata survive distribution shift because they encode role/invariant structure, not just surface regularities.
- **Intervention predictiveness:** If a person-level update (“renegotiate terms to make scope explicit”) commutes with a system-level change (“tighten access controls”), we can predict which combination reduces breach risk without trial-and-error.
- **Auditability:** Commutation exposes where explanations break: a non-commuting square tells us whether the failure is in abstraction (bad T), in implementation (bad S), or in the layer-specific explanans (bad E_k or E_{k+1}).

Minimal metric:

- **Commutation error (CE):** CE = d(T \circ E_k \circ I , E_{k+1} \circ T \circ I), with d chosen per domain (e.g., TV distance, policy divergence, semantic discrepancy).
- **Role stability (RS):** Fraction of interventions for which role assignments (constraints/dynamics/sources) are invariant under T and S. Low RS signals hidden reduction or ad hoc patching.

What “fit” misses:

- A high-AUC classifier of cooperative vs. defecting episodes may fail when incentives change, because it never encoded the **promise** invariant. A commuting explanation, by contrast, ensures that the person-level predicate (kept promise) aligns with the system-level mechanisms that realize it across changes.

Design heuristic:

- Prefer explanations whose diagrams commute over those with marginally better in-grain fit. When both are available, pick the one with lower CE even at small cost in raw loss; expect better OOD performance and safer policy transfer.

In sum, moving **beyond nothing-but** means: keep mechanisms, restore norms; require that bridging maps preserve the higher-level content under intervention; and judge explanations by whether their diagrams commute, not only by how well they fit a single dataset.

2. Principles Without Reduction

2.1 Constraint-first causation: formal and final causes alongside efficient causes

Claim. Causation is not exhausted by efficient pushes and pulls. Two additional, empirically tractable kinds matter in science:

- **Formal causes (structure):** the admissible-shape constraints that carve the space of trajectories (e.g., conservation laws, symmetries, network topology, interface specifications).
- **Final causes (ends):** target states/invariants that policies or organisms are organized to preserve or realize (homeostasis, promise-keeping, mission goals).

Operationalization.

- Let X be a state space, F a flow (efficient cause), C a constraint set, and G a goal/invariant set.
- Dynamics with constraints: x_{t+1} = F(x_t, u_t) subject to x_t in C.
- Goal-conditioned viability: a policy pi is *G-viable* on horizon H if all induced trajectories satisfy Inv_G(x_{t:t+H}) = 1.

Testable upshot.

1. Adding correctly identified (C,G) shrinks hypothesis space and improves out-of-distribution (OOD) generalization without changing raw micro-laws.
2. Interventions that preserve (C,G) deliver effects predicted at higher strata (life, mind, person) more reliably than interventions that only push F.

Falsifier. If models that include formal/final constraints do *not* deliver better intervention predictions or OOD performance than efficient-only baselines at equal capacity, the constraint-first thesis is weakened.

Examples.

- **Biology:** Stoichiometry and topology (formal) + homeostatic setpoints (final) predict which gene knockouts are survivable.
- **Cognition:** Grammar constraints (formal) + communicative intent (final) explain repair strategies beyond n-gram frequency.
- **Alignment:** Protocol invariants (formal) + truth-telling/promise-keeping (final) predict cooperative stability beyond reward hacking.

2.2 Roles over parts: constraints, dynamics, sources (CDS)

Idea. Explanations should identify *roles* that remain stable across implementations, not just list parts. We use three roles:

- **C: Constraints** — what must be preserved; admissible region; guardrails.
- **D: Dynamics** — how state updates occur within constraints; mechanisms of change.
- **S: Sources** — exogenous drives, resources, and boundary supplies (inputs, incentives, affordances).

Minimal schema.

- System M is explained by a triple (C, D, S).
- Interventions are typed by the role they target: I_C (tighten/relax admissibility), I_D (modify update rules), I_S (alter supplies/drives).

Commutation requirement (role-aware).
We demand role-stable explanations across strata via maps T (up) and S_lift (down):

- Role preservation (informal): T(C,D,S) -> (C',D',S') with roles intact.
- Intervention commutation (ASCII-safe):
  T o E(C,D,S) o I_r ~= E'(T(C,D,S)) o T o I_r, for r in {C,D,S}.

Why roles beat parts.

- Two systems with different micro-parts but the same CDS roles can be substituted in a process without loss of function (principle of *role equivalence*).
- Role annotations make failure *localizable*: breaches of C, drift in D, or starvation at S produce distinct, predictable signatures.

How to learn CDS from data.

1. Fit candidate mechanisms.
2. Infer minimal C that renders D well-posed and predictive.
3. Partition exogenous influences into S and test via counterfactuals.
4. Validate by role-targeted interventions: does an I_C move behave like a constraint edit across implementations?

Metrics.

- **Role stability (RS):** fraction of interventions whose predicted effects are invariant under T,S_lift.
- **Commutation error by role (CE_C, CE_D, CE_S):** discrepancy when executing I_r before vs after crossing strata.
- **Substitution score (SS):** OOD performance when swapping different micro-implementations that share CDS roles.

Worked micro-examples.

- **Anesthesia:** C = viability bounds for neural coordination; D = channel/network dynamics; S = anesthetic concentration as drive on receptors. Inducing I_S reduces coordination without redefining C; recovery re-enters C's viable region.
- **Contract compliance:** C = policy invariants; D = workflow transitions; S = incentives/penalties. Tightening C before raising S prevents reward-only gaming.

2.3 Conservativity: extending without contradiction

Goal. Let higher-level theories add evaluative and teleological structure while remaining *conservative* over established lower-level laws.

Definition (informal, category-theoretic spirit).

- Let L be a lower-layer theory (physics/biophysics) and H a higher-layer extension (life/mind/person) with an embedding functor J: L -> H.
- H is a *conservative extension* of L if every sentence phi in the language of L that is provable in H is already provable in L. Intuition: H adds structure (new predicates, invariants, norms) but no new theorems purely about L's primitives.

Engineering rule.

- Add structure as constraints/invariants on admissible models, not as new forces violating L.
- Use *refinement types*: x:State where Inv_H(x) to express higher-level admissibility.

Why this matters.

- Prevents "modal collapse" (everything necessary) and physics violations.

- Keeps metaphysical commitments from doing hidden empirical work.
- Enables shared audit: physicist checks L-claims, ethicist/person-level analyst checks H-claims, and the bridge laws specify how they meet.

Proving/assessing conservativity (pragmatic).

1. **Syntactic route:** Give H as L + axioms over new symbols; show no new theorems in the old alphabet.
2. **Semantic route:** Build a forgetful functor U: Mod(H) -> Mod(L) that is surjective-on-objects and preserves satisfaction of L-sentences.
3. **Empirical guardrail:** Any H-guided intervention must compile into L-admissible manipulations; if an H-prediction requires an L-violation, the method is out of bounds.

Patterned examples.

- **Biology:** Add metabolic invariants (Inv_H) to L = chemical kinetics. H predicts which interventions are viable; it does not alter reaction laws.
- **Mind:** Add predicates for correctness/attention to dynamical neural models; predictions compile to admissible perturbations (stimulation, pharmacology) without new "mental forces."
- **Person/Covenant:** Add promise-keeping and fiduciary duties as constraints on policy sets; compiled actions are access controls, audits, and incentives compatible with system dynamics.

Falsifier. If the higher-layer account repeatedly yields accurate predictions *only* by appealing to effects that cannot be compiled into L-admissible interventions, H is not conservative and should be revised or rejected.

Design checklist (non-reductive, conservative).

- Specify CDS roles at each stratum.
- State higher-layer invariants Inv_H explicitly.
- Provide T (up) and S_lift (down) with commutation tests.
- Exhibit U (forgetful) and/or a syntactic argument that H is conservative over L.
- Pre-register role-targeted interventions and OOD tests.
- Fail gracefully: report which square fails to commute (bad T, bad S_lift, bad E).

Payoff. We keep the reach of science (mechanism, prediction) while admitting ends and norms where they actually govern. Explanations travel across strata because they preserve roles and constraints, and they do so without smuggling in physics-breaking claims.

3. Mathematics of Bridging, Not Flattening

3.1 Explanations as functors; naturality as commutation

Categories.
For each stratum k (physics, life, mind, person, covenant) let Cat_k be a category:

- Obj(Cat_k): systems or theories at layer k
- Mor(Cat_k): structure-preserving maps (model homomorphisms, simulators, policy refinements)

Let Int_k be a category of **interventions** at layer k:

- Obj(Int_k): admissible intervention contexts (actuator sets, protocols)
- Mor(Int_k): refinement/combination of interventions

Explanans as functor.
An explanation at layer k is a functor

- E_k : Int_k -> Pred_k
  where Pred_k is a category of predictions/evaluations at layer k (distributions, invariants, judgments).

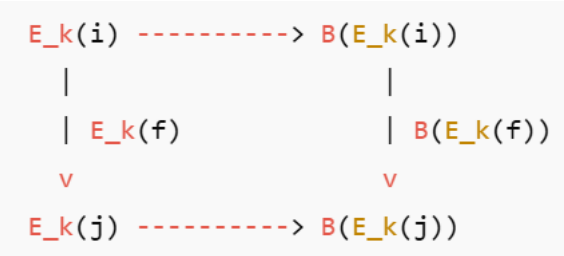
Bridging functors.
Between layers k -> k+1 we give two structure-preserving maps:

- T_{k->k+1} : Cat_k -> Cat_{k+1} (coarse-grain / abstraction)
- B_{k->k+1} : Pred_k -> Pred_{k+1} (prediction lift across vocabularies)

Naturality (commutation).
A family of maps eta: B o E_k => E_{k+1} o A is a **natural transformation** between functors

- A : Int_k -> Int_{k+1} (how a k-level intervention is seen above)
- B : Pred_k -> Pred_{k+1} (how k-level predictions are read above)

The **naturality square** for each i in Int_k should commute:



and likewise with A and E_{k+1}. We measure **commutation error**:

- CE = d(B(E_k(i)), E_{k+1}(A(i))), with domain-appropriate metric d.

Reading.
Explanations *travel* when eta is (approximately) natural: whether we explain before or after lifting across layers, the result agrees within pre-registered tolerance.

3.2 Adjoint pairs for coarse-grain and lift (upward/downward guidance)

We model up- and down-bridging with an adjunction:

- CG : Cat_k -> Cat_{k+1} (coarse-grain, abstraction)
- LF : Cat_{k+1} -> Cat_k (lift, implementation/guidance)
- LF ⊣ CG (LF is left adjoint to CG)

Adjunction data.

- Unit: eta : Id_{Cat_{k+1}} -> CG o LF
- Counit: eps : LF o CG -> Id_{Cat_k}
  satisfying triangle identities:
- CG(eps) o eta_CG = id_CG
- eps_LF o LF(eta) = id_LF

Intuition.

- CG forgets micro-detail but preserves the higher-level roles/invariants needed above.
- LF implements higher-level specifications as admissible lower-level configurations.

- The adjunction guarantees **best possible** alignment: LF(x^) is the most liberal k-level realization of an (k+1)-level spec x^; CG(y) is the tightest (k+1)-level summary of a k-level system y.

Order-theoretic sanity check.
In posets (monotone maps), an adjunction is a Galois connection:

- LF(x^) <= y iff x^ <= CG(y)
  This captures **downward guidance without physics violation**: high-level admissibility x^ constrains which y are allowable; mechanisms remain within Cat_k.

Role-aware adjunction.
Let each object carry **CDS** roles (Constraints, Dynamics, Sources). Require:

- CG preserves roles: CG(C,D,S) -> (C^,D^,S^) with role labels intact.
- LF respects roles: LF(C^,D^,S^) returns implementations that realize C^ as invariants, D^ as update rules, S^ as supplies.

Error budgets.

- Implementation loss: L_impl = dist(CG(LF(x^)), x^)
- Abstraction loss: L_abs = dist(y, LF(CG(y))) (measures over-/under-fitting of summaries)

Pre-register tolerances for both losses; adjoint design aims to minimize them.

3.3 Sheaf semantics for multi-scale coherence and gluing tests

Sites and covers.
Fix a domain X (system, organization, organism). Let Open(X) be a cover system (subsystems, times, modalities). A **presheaf** F assigns to each U in Open(X) a set/object F(U) of local models/predictions, and to each inclusion V<=U a **restriction** map res^U_V : F(U) -> F(V).

Sheaf condition (gluing).
Given a cover {U_i} of U and a family of local sections s_i in F(U_i) such that

- res^{U_i}*{U_i \cap U_j}(s_i) = res^{U_j}*{U_i \cap U_j}(s_j) for all i,j
  there exists a unique s in F(U) with res^U_{U_i}(s) = s_i for all i.

Interpretation.

- Local stories agree on overlaps => there exists a global story.
- Failure to glue quantifies **incoherence across scales or modalities**.

Our use.

- Let Sh_k be a sheaf of k-level explanations/predictions over a cover by scales (micro, meso, macro) or views (mechanistic, behavioral, normative).
- Compute **Cech inconsistency** on overlaps as a gluing obstruction:
    - GI = aggregate discrepancy on pairwise overlaps
    - If GI > tau, there is no global section; the model fails to unify the local accounts.

Role-aware sheaves.
Decompose into role-sheaves:

- Sh_C, Sh_D, Sh_S with their own restriction maps.
- A composite explanation glues only if each role glues (stronger test than gluing the aggregate).

Algorithmic test (pragmatic).

1. Choose a finite cover {U_i}.
2. Fit local models s_i in each U_i with CDS annotation.
3. Evaluate overlap errors e_{ij} = d(res(s_i), res(s_j)).
4. Solve a global section fit (consistency projection) or prove non-existence within tolerance.
5. Report which role-sheaf fails (C vs D vs S) to localize the mismatch.

Payoff.

- Early warning: rising GI often precedes KPI drift.
- OOD robustness: models that pass gluing on historical covers transfer better to novel partitions.

3.4 Copy-safe math conventions and notation (ASCII-safe)

To keep the paper paste-ready for Word/plain text, we adopt the following **ASCII-safe** conventions:

Symbols and composition.

- Function application: f(x), composition: g o f, identity: id_X

- Functors: F: C -> D
- Natural transformation: eta: F => G (components eta_X: F(X) -> G(X))
- Adjunction: LF ⊣ CG with unit eta: Id -> CG o LF and counit eps: LF o CG -> Id
- Products, pairs: (x,y); tuples: (x1,...,xn)
- Equality up to tolerance: ~= (numerically), === (definition)

Sets, categories, structures.

- Categories named Cat_k, Int_k, Pred_k
- Objects, morphisms: Obj(C), Mor(C)
- Presheaf/Sheaf: F, Sh; restriction res^U_V
- Covers: {U_i}, overlaps: U_i cap U_j

Roles and interventions.

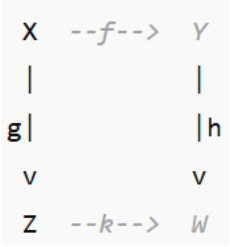
- Roles: C (Constraints), D (Dynamics), S (Sources)
- Role-targeted intervention: I_C, I_D, I_S
- Up- and down-bridging: T (or CG) for coarse-grain, S_lift (or LF) for lift

Losses and metrics.

- Commutation error: CE = d(B(E_k(i)), E_{k+1}(A(i)))
- Role-stability: RS in [0,1]
- Implementation loss: L_impl; Abstraction loss: L_abs
- Gluing index: GI; overlap errors: e_{ij}

Diagrams (ASCII).

- Commuting square:



- Naturality (component-wise) means h o eta_X = eta_Y o F(f).

Quantification and proofs.

- Use explicit quantifiers in prose: “for all i in Int_k”, “there exists s in F(U)”
- No Greek letters required; if needed, write alpha, beta, gamma in ASCII.

Copy-safety checklist.

- No Unicode math glyphs; use ASCII only.
- No specialized equation editors; equations render in plain text.
- Diagrams drawn with ASCII boxes/arrows.
- Variables and functors consistently named across sections.
- All metrics defined with typed ranges and units where applicable.

Mini worked toy (pulling 3.1–3.3 together).

- k = life, k+1 = mind.
- CG abstracts spike trains to symbolic acts; LF realizes tasks as admissible neural policies.
- E_k predicts viability under I_S (neurotransmitter modulation); E_{k+1} predicts reportable contents under A(I_S).
- We compute CE across a set of pharmacological interventions; we compute GI across a cover by tasks {perception, recall, decision}.
- Acceptance: CE <= tau_CE, GI <= tau_GI, RS >= tau_RS.
  Passing these implies the explanation *travels* (functorial/natural), can be *guided down* without physics violation (adjoint), and *glues* across tasks/scales (sheaf) — bridging, not flattening.

4. The Role Calculus (CDS)

4.1 Definitions: constraint sets, dynamical laws, source terms

System tuple.
A modeled system on horizon H is a triple M = (C, D, S) over state space X and action space U.

- **Constraint set C.**
  C is a family of admissibility predicates on trajectories and updates.

- State admissibility: Adm_X(x_t) in {0,1}.
- Transition admissibility: Adm_T(x_t,u_t,x_{t+1}) in {0,1}.
- Invariants: collection Inv = {I_k} where each I_k is a predicate on paths x_{t:t+h} that must hold (e.g., conservation, policy clauses, safety).
- Write C = (Adm_X, Adm_T, Inv).

- **Dynamical law D.**
  D specifies how states update *within* C.
  - Deterministic: x_{t+1} = f(x_t, u_t; theta).
  - Stochastic: x_{t+1} ~ P_theta(. | x_t, u_t).
  - Learned policy/environment pairs are allowed; D can factor as (f_env, pi) or (P_env, pi).

- **Source terms S.**
  Exogenous drives/supplies that impinge on D but are not decided by D.
  - Inputs: w_t (disturbances, market shocks, anesthetic dose).
  - Budgets: resource bounds R (time, energy, cash).
  - Affordances/incentives: payoff schedules, penalties.
  - Write S = (W, R, Phi) where W = {w_t}, R are bounds, Phi encodes incentives.

Well-posedness (role sanity).
Given M = (C, D, S), an episode is *valid* if for all t:

1. Adm_X(x_t) = 1.
2. x_{t+1} sampled/generated by D satisfies Adm_T(x_t,u_t,x_{t+1}) = 1.
3. For all invariants I_k and windows h in H, I_k(x_{t:t+h}) = 1.
4. Source usage respects budgets R.

Role minimality.
A role assignment is *minimal* if removing any element from C or S, or weakening D, either violates validity on observed episodes or worsens pre-registered predictive criteria beyond tolerance. (MDL-style: shortest description that still works.)

Example sketches.

- Biochemical network: C = stoichiometry + bounds; D = mass-action kinetics; S = nutrient inflows.
- Contract workflow: C = policy invariants; D = permitted state transitions; S = incentives, audits.
- Anesthesia model: C = viability band for neural coordination; D = network/channel dynamics; S = dose-time curve.

4.2 Intervention semantics and commutation error

Typed interventions.
Interventions are edits to exactly one role at a time (atomic), or compositions thereof.

- Constraint edit I_C: tighten/relax admissibility or add/remove invariants.
  Example: add “two-person rule” invariant to a process; add a safety bound to voltage.
- Dynamics edit I_D: alter update law while keeping C fixed.
  Example: change controller gain; swap a policy network; ablate a circuit.
- Source edit I_S: change exogenous drives/supplies/incentives.
  Example: increase anesthetic concentration; alter payoff for defection.

Semantics.
Given M = (C, D, S) and an atomic I_r, define

- I_C(M) = (C', D, S)
- I_D(M) = (C, D', S)
- I_S(M) = (C, D, S')
  Compositions respect role order when specified; otherwise default to left-to-right application.

Across-strata maps.

- Up-map (coarse-grain) T: M_k -> M_{k+1}.
- Down-map (lift) L: M_{k+1} -> M_k.
  Both are role-aware: T(C,D,S) = (C^,D^,S^) and similarly for L.

Commutation error (role-specific).
We require that “explain then lift” and “lift then explain” give nearly the same effect.

- Predictive functors: E_k, E_{k+1} at layers k and k+1.
- For an intervention I_r at layer k, define

CE_r(k->k+1; I_r) = d(B(E_k(I_r(M_k))), E_{k+1}(A(I_r)(T(M_k))))

where
A(I_r) is the induced intervention at layer k+1,
B lifts predictions k->k+1, and
d is a domain metric (TV distance, policy KL, violation rate gap, etc.).

Aggregate commutation error (ACE).

ACE = w_C * E_I[CE_C] + w_D * E_I[CE_D] + w_S * E_I[CE_S]

with weights w_r summing to 1 and expectation over a pre-registered intervention set.

Pass/fail criterion.
An explanation *travels* if ACE <= tau and each role-wise mean error E_I[CE_r] <= tau_r.

Why this matters.

- Low CE_C means higher-level constraints reflect genuine admissibility, not post-hoc regularizers.
- Low CE_D means mechanism claims survive abstraction.
- Low CE_S means we distinguished endogenous dynamics from exogenous drives correctly.

Minimal intervention suite.
For identifiability (next subsection) we need a separating set:

- At least one I_C that changes feasibility without altering D or S.
- At least one I_D that changes update behavior while satisfying all original invariants.
- At least one I_S that alters exogenous drives without changing feasibility sets.

4.3 Identifiability: role stability across model classes

Goal.
Show when roles (C,D,S) are *recoverable* from data and remain stable if we swap micro-implementations (different parts, same roles).

Equivalence by role.
Two models M and M' are role-equivalent, written M ~_role M', if:

1. They share the same constraints up to monotone equivalence: C ≡ C' (admit the same trajectories).
2. They induce observationally indistinguishable dynamics inside C under the same sources S (up to tolerance on predictions).
3. They expose the same source-response map: for any admissible I_S, the induced policy/outcome distributions match up to tolerance.

Role stability (RS).
Let I = I_C ∪ I_D ∪ I_S be a test battery. Define

RS = 1 - E_{i in I}[1(predicts differ under role-equivalent swap)]

RS close to 1 means predictions are invariant under substitution of micro-implementations that preserve roles.

Identifiability conditions (informal but testable).

- **IC1 (separation by effects).**
  There exists a battery of typed interventions such that each role produces a *unique signature*:
  - I_C: feasibility boundary shifts without change to within-boundary local response.
  - I_D: response kernel changes within the same feasibility region.
  - I_S: distribution over exogenous variables changes with no boundary shift.
- **IC2 (rich overlap).**
  Data include episodes that probe neighborhoods just inside and just outside constraint boundaries, plus multiple source regimes for the same boundary and dynamics.
- **IC3 (no perfect confounding).**
  There is no policy or environment that simultaneously mimics a constraint-tightening and a source shift for all probes in the battery. (Design batteries to break such mimicry.)
- **IC4 (minimality/parsimonious roles).**
  Among all role assignments that fit the data within tolerance, choose the one with minimal role description length:
- MDL_role = DL(C) + DL(D) + DL(S)

Penalize leakage (explaining constraint violations by source tweaks or vice versa).

Learning procedure (sketch).

1. **Fit candidates.** Train a family of mechanistic/policy models that explain observed transitions.
2. **Infer constraints.** Solve for the weakest C that renders D well-posed and matches violation statistics. (Can be cast as max-margin over trajectories or as differentiable barrier inference.)
3. **Separate sources.** Decompose residual exogenous variation into S (inputs, incentives) using instrumental variables or regime switches.
4. **Audit roles.** Run the intervention battery in simulation or on held-out regimes; compute CE_C, CE_D, CE_S and RS.
5. **Select by MDL_role + ACE.** Choose the role assignment minimizing MDL_role subject to ACE <= tau and RS >= tau_RS.
6. **Stress-swap.** Replace micro-implementation (e.g., swap a controller, use another architecture) keeping inferred roles; re-run audits to verify role stability.

Diagnostics when identifiability fails.

- **High CE_C only:** constraints are misspecified; tighten/reshape C (wrong invariants or boundaries).
- **High CE_D only:** mechanism is underfit/overfit; add state, fix causality, or refactor policy.
- **High CE_S only:** source leakage; your "exogenous" factors are partly endogenous—re-scope.
- **Low RS with low ACE:** roles are fragile under swap; refine role definitions to be implementation-invariant (you are still keying on parts).
- **Battery non-separating:** add probes that isolate each role (e.g., a pure budget shock for S, a pure guardrail edit for C).

Concrete toy (copy-safe).

- Queueing system with service policy.
    - C: max queue length L, no negative inventory, fairness invariant on wait time.
    - D: service rule (FIFO vs SRPT) and arrival/service kernels.

    - S: arrival rate lambda, burst process B_t, staffing budget R.

- Interventions:
    - I_C: set L' < L (tighten), fairness on/off.
    - I_D: switch SRPT->FIFO.
    - I_S: double lambda for 1 hour.
- Identifiability:
    - If tightening L changes overflow frequency but not per-customer service-time distribution inside L, that is a C-signature.
    - If SRPT->FIFO alters wait distribution without changing overflow at same L and lambda, D-signature.
    - If doubling lambda increases wait and overflow without changing fairness violations (when on), S-signature.
- Role stability:
    - Swap SRPT implementation between two codebases; if predictions under the same (C,S) match within tolerance, RS remains high.

Takeaway.
CDS turns “what is going on here?” into a falsifiable partition: what must hold (C), how change proceeds (D), what drives it (S). With typed interventions, commutation tests, and role-stability audits, the partition becomes identifiable and portable across implementations—exactly what we need for explanations that travel rather than dissolve.

5. Logos-Interpreter (Benchmark)

5.1 Tasks: diagram-commuting explanations under interventions

Goal. Evaluate whether explanations *travel* across strata and interventions by testing commutation and role stability (CDS).

Task families (each with typed intervention suites I_C, I_D, I_S):

1. **Bio-control (chemostat/GRN).**
    - Layers: physics/biochem (k) -> life/regulatory (k+1).
    - Inputs: gene circuits, nutrient flows, knockout/overexpression edits.

- Queries: predict viability windows, set-point recovery, off-target effects.
- Interventions: I_C (add safety bound on metabolite), I_D (swap kinetic law or controller), I_S (change inflow profile).

2. **Cognitive agents (gridworld->dialogue).**
    - Layers: policy dynamics (k) -> person-level norms (k+1).
    - Inputs: tasks with explicit promises/obligations.
    - Queries: will the agent keep a stated promise under incentive shifts?
    - Interventions: I_C (tighten policy invariant: “no lying”), I_D (swap planner/architecture), I_S (alter rewards/penalties).
3. **Socio-technical workflow (access control).**
    - Layers: system transitions (k) -> covenant/policy compliance (k+1).
    - Inputs: ticketing states, role hierarchies, audit rules.
    - Queries: breach risk under policy edits or incentive changes.
    - Interventions: I_C (two-person rule on sensitive ops), I_D (change approval automaton), I_S (introduce bonus for speed).
4. **Robotics (dynamics->task specification).**
    - Layers: low-level control (k) -> task/goal spec (k+1).
    - Inputs: plant models, constraints (joint limits, safety zones).
    - Queries: success/failure under path changes and constraint edits.
    - Interventions: I_C (narrow safety corridor), I_D (controller retune), I_S (disturbance injection).

Per-task deliverables.

- A role-annotated dataset D with (C,D,S), interventions, outcomes.
- Bridges T (coarse-grain) and L (lift) with documentation.
- Gold/reference explanations E_*k, E*_{k+1} for audit calibration.

5.2 Metrics: commutation error, OOD generalization, auditability

Core commutation metrics.

- **CE_r (role-specific):**
  CE_r = d(B(E_k(I_r(M_k))), E_{k+1}(A(I_r)(T(M_k)))), r in {C,D,S}.
  d = task-appropriate metric (e.g., TV distance on distributions, policy KL, invariant violation rate gap).
- **ACE (aggregate):**
  ACE = w_C * E[CE_C] + w_D * E[CE_D] + w_S * E[CE_S], weights sum to 1.
- **RS (role stability):** invariance of predictions under role-equivalent micro-swaps.

Generalization metrics.

- **CE-OOD:** CE_r on interventions/regimes unseen in training (distribution, mechanism, or cover shift).
- **Transfer-commute score (TCS):** fraction of novel tasks for which ACE <= tau and RS >= tau_RS.
- **Gluing index (GI):** sheaf inconsistency across covers (scales/views); lower is better.

Auditability metrics.

- **Explanation provenance completeness (EPC):** fraction of claims linked to artifacts (T,L, role maps, tests).
- **Intervention traceability (IT):** percent of predictions with a reproducible intervention script.
- **Diagram coverage (DC):** percent of required naturality squares instantiated and evaluated.
- **Counterfactual locality (CL):** fraction of counterfactuals whose effects are confined to the targeted role (sanity for typed interventions).

Pass thresholds (per track).

- Minimal: ACE <= tau_min, RS >= tau_RS_min, GI <= tau_GI.
- Strong: CE_r <= tau_r_strong for all r; CE-OOD within 1.2x in-domain; DC >= 0.9; IT >= 0.95.

5.3 Baselines and ablations

Baselines (to beat).

1. **Pure fit (reductionist).** Single-grain black-box predictor (e.g., deep policy/value net) trained to minimize in-grain loss; no roles, no bridges.
2. **Post-hoc explainer.** LIME/SHAP-style feature attributions layered after a black box; no role typing; evaluates only within-grain.
3. **Constraint-only.** Hard-coded guardrails without mechanism/source differentiation (all errors blamed on C).
4. **Reward-only alignment.** No truth/promise constraints; learns to game incentives.

Our model classes.

- **Role-typed models:** learn (C,D,S) explicitly; bridges T,L provided or learned with constraints.
- **Adjoint-regularized models:** penalties on abstraction/implementation losses (L_abs, L_impl).
- **Sheaf-consistent ensembles:** local models with gluing projection.

Ablations (to locate gains).

- **-Roles:** remove CDS typing; keep same capacity. Expect CE_r jumps, RS drops.
- **-Adjoint:** replace LF ⊣ CG with unconstrained up/down maps. Expect higher L_abs/L_impl, worse CE-OOD.
- **-Sheaf:** no gluing penalties; expect GI spikes and brittle transfer.
- **-Final causes:** drop goal/invariant G from C; expect failure under I_S shifts.
- **-Formal causes:** drop structural constraints; expect mechanistic overfit and poor RS.
- **+Noise bridges:** perturb T,L to test robustness; measure change in ACE and RS.

Reporting.

- Learning curves for ACE and GI.
- Pareto plots: (ACE, CE-OOD, RS) vs capacity.
- Role confusion matrices under the intervention battery.

5.4 Failure modes and diagnostic reports

Canonical failure modes.

1. **Constraint leakage.** Effects that should be attributed to C appear as D or S.
    - Symptom: high CE_C, low CE_D/CE_S; boundary probes mispredicted.
    - Fix: refine C (shape, thresholds), strengthen role regularization.
2. **Mechanism myopia.** Model fits invariants but misses update structure.
    - Symptom: high CE_D; OOD rollouts diverge though constraints hold.
    - Fix: enrich state, causal factorization, identify hidden dynamics.
3. **Source confounding.** Exogenous/endogenous mixed.
    - Symptom: high CE_S; interventions on S look like constraint edits.
    - Fix: re-scope S with instruments/regime switches.
4. **Bridge breakage.** Up/down maps not faithful.
    - Symptom: large L_abs or L_impl; ACE rises despite good in-grain fit.
    - Fix: retrain T,L with adjoint regularization; redesign abstractions.
5. **Non-gluing locals.** Local models good; global story impossible.
    - Symptom: GI >> tau; overlap residuals e_{ij} cluster by role.
    - Fix: revise cover or reconcile locals; add cross-view constraints.
6. **Role instability under swap.** Explanations depend on parts, not roles.
    - Symptom: RS < tau_RS with low ACE in-domain.
    - Fix: re-define roles at correct abstraction; penalize part-specific features.

Diagnostic report template (ASCII-safe).

LOGOS-INTERPRETER DIAGNOSTIC

Task: <name> Seed: <id> Version: <git-hash>

Summary:

ACE=0.073 (tau=0.10) | RS=0.92 (tau_RS=0.90) | GI=0.028 (tau_GI=0.05)

CE_C=0.041 | CE_D=0.026 | CE_S=0.019

CE-OOD=0.085 | DC=0.94 | IT=0.98 | EPC=0.96

Role Confusion (battery B):

true\pred	C	D	S
C	.91	.06	.03
D	.07	.88	.05
S	.04	.05	.91

Bridge Losses:

L_abs=0.017 | L_impl=0.022

Overlap Errors (top-5):

e_{U1,U2}=0.031 [role=C], e_{U2,U3}=0.018 [role=D], ...

Alerts:

- Mild constraint leakage near boundary x~0.7 (CE_C spike).
- Non-gluing at U1 cap U2; recommend cover refinement or added invariant I_3.

Actions:

- Tighten C: add barrier on feature phi_7.
- Retrain T,L with adjoint weight lambda=0.5 -> 0.8.
- Add I_S^pure (budget-only shock) to separate S from C in B'.

Leaderboard protocol.

- Submit predictions + artifacts (T,L, role maps, intervention scripts).
- Repro check: IT >= 0.95 required for ranking.
- Primary ordering by ACE; ties broken by CE-OOD, then RS, then GI.
- Separate tracks: Simulated, Real-world (restricted), Cross-domain transfer.

Takeaway.
Logos-Interpreter rewards explanations that *commute, glue, and persist under swap*. It is not an essay contest: success requires artifacts that preserve roles (C,D,S), minimize commutation error in and out of distribution, and expose where the story fails so it can be repaired.

6. Teleology-Finder (Tool)

6.1 Inferring ends: extracting purpose-consistent constraints

Goal. Given trajectories, interventions, and outcomes, infer the *ends* (targets/invariants) a system is organized to preserve, and express them as constraint predicates that can guide prediction and control.

Inputs.

- Trajectory set T = { tau_i }, each tau = (x_0,u_0,...,x_H).
- Typed interventions B = I_C ∪ I_D ∪ I_S with execution logs.
- Candidate feature library Phi = { phi_j(x,u) } (can include learned embeddings).
- Tolerances eps_inv (violation), eps_pred (fit), and complexity budgets.

Outputs.

- End set G = { I_k } where each I_k is an invariant predicate on paths or eventals:
    - Pointwise: I_k(x_t) = 1.
    - Temporal: I_k(x_{t:t+h}) = 1 (e.g., dwell-time, bounded accumulation).
    - Relational: I_k(f1,...,fm) = 1 (e.g., fairness across agents).
- Role update C := (Adm_X, Adm_T, Inv ∪ G) with certificates.

Problem form (max-margin invariants).
Find the smallest family G such that

for all tau in T_positives: I_k(tau) = 1 for all k

for all tau' in T_counterf: exists k s.t. I_k(tau') = 0

minimize DL(G) subject to violation rate <= eps_inv

where T_counterf are counterfactual or near-miss trajectories gathered from B.

Construction recipes.

- **Barrier learning (continuous domains).**
    - Learn g(x) >= 0 with parameterized g s.t. along valid trajectories
      g(x_t) >= 0 and g(x_{t+1}) >= 0, while maximum margin to invalids.
    - Enforce discrete Nagumo-like condition:
      g(x_t) >= 0 and Adm_T => g(x_{t+1}) >= -delta.
- **Temporal invariants (spec mining).**
    - Use syntax-guided search over safety LTL fragments:
      G(p), G(q -> F_{<=h} r), G(x in [a,b]).
    - Score by (1) coverage of positives, (2) exclusion of counterfactuals, (3) MDL on formula.
- **Relational ends (multi-agent).**
    - Infer constraints over aggregates, e.g.,
      |avg_reward_i - avg_reward_j| <= eta,
      median wait-time parity, or no-negative-transfer:
      G(perf_groupA >= perf_groupB - eta).

Disentangling ends from mere regularities.

- Require robustness under I_D and I_S: an end remains predictive across planner swaps and supply shocks.
- Reject propositions that evaporate when D or S changes mildly.

Certificates.

- For each I_k, report:
    - Training/validation violation rates.

- Margin histograms w.r.t. counterfactuals.
- Role-specific commutation deltas ΔCE_r when adding I_k.

6.2 From learned models to minimal end-congruent programs

Goal. Starting from a trained mechanism (policy/model), synthesize a *program of ends* (constraints and controllers) that realizes the same successes with fewer degrees of freedom and clearer invariants.

Pipeline.

1. **Mechanism distillation.**
   Fit a surrogate D_hat that is simulable and differentiable (or an interpretable policy automaton).
2. **End mining.**
   Run 6.1 to get G = { I_k }. Prune by MDL and stability across interventions.
3. **Controller synthesis under ends.**
   Compute a minimal controller pi_min so that trajectories under (C ∪ G, D_env, pi_min, S) satisfy goals within eps_pred.
   - Continuous: constrained MPC with terminal set from barrier g(x) >= 0.
   - Discrete: supervisor synthesis (winning set in safety game).
4. **Program extraction.**
   Emit a human-auditable spec:
5. ENDS:
6. I1: g1(x) >= 0 (safety envelope)
7. I2: G(stock in [a,b]) (setpoint band)
8. I3: fairness <= eta (relational)
9. CONTROLLER:
10. u_t = K(x_t) if g1(x_t) > tau else u_safe
11. ...
12. SOURCES:

13. budgets R, admissible disturbances W
14. **Refinement loop.**
    If fit deviates: update G (add/remove), or expand controller class until ACE and GI pass.

Objective.

minimize DL(G) + DL(controller)

subject to ACE <= tau, RS >= tau_RS, GI <= tau_GI

violation_rate(G) <= eps_inv

task_loss <= eps_pred

Auditable artifacts.

- Source code for ends and controller.
- Intervention scripts and seed list.
- Bridge maps T,L used for commutation checks.

6.3 Case studies: biology, cognition, cooperative agents

Biology: homeostatic metabolism (chemostat).

- Data: nutrient inflow variations, knockout episodes, recovery traces.
- Ends found:
  - I1: G(biomass in [b_min,b_max])
  - I2: G(toxic_metabolite <= tau)
  - I3: G(NADH/NAD+ ratio in [r_min,r_max])
- Program: MPC that enforces I1–I3; safety supervisor that throttles feed rate when g_toxic approaches 0.
- Results: lower CE_S under unseen inflow shocks; earlier fault detection (GI drop).

Cognition: dialog agent with truth/promise invariants.

- Data: tasks with explicit commitments plus adversarial incentives.
- Ends found:
  - I1: no_contradiction(history ∪ output)

    - I2: keep(commitments, horizon <= H)
    - I3: cite(if knowledge_claim)
- Program: decoding with constraint projection (repair search) and a lightweight verifier; refusal policy when I1 cannot be satisfied.
- Results: improved OOD honesty; ACE falls sharply under I_S (reward perturbations); RS high across model architectures.

Cooperative agents: access-control workflow.

- Data: ticket flows, role changes, breach incidents, incentive tweaks.
- Ends found:
    - I1: two_person_rule on op ∈ S_sensitive
    - I2: separation_of_duties(r1,r2)
    - I3: SLA_fairness <= eta across teams
- Program: automaton-level supervisor; incentive redesign consistent with ends; compiled to system policies.
- Results: breach risk predicted and reduced without new rules at the micro level; low CE_C across policy edits.

6.4 Guarantees and limits

Guarantees (under assumptions).

- **End stability.** If an invariant I_k passes robustness tests under separating batteries B (IC1–IC3) and MDL_role selection, then for any role-equivalent implementation M' ~_role M, the violation probability of I_k differs by at most delta (uniform generalization bound from VC/Rademacher of chosen spec class).
- **Conservativity.** Synthesis compiles to admissible low-level actions; no new theorems about the lower layer are introduced (semantic conservativity via forgetful functor U).
- **Safety envelope soundness.** For barrier g(x), if learned with non-negative margin and enforced with a controller that satisfies the discrete Nagumo condition, trajectories remain in { x : g(x) >= -delta } up to discretization error.

Limits.

- **Spec expressivity.** Ends outside the grammar (e.g., non-Markov relational obligations with unobserved context) may not be captured; GI flags non-gluing but cannot invent missing observables.
- **Confounding.** If data lack a separating intervention battery (IC1–IC3 fail), Teleology-Finder can overfit regularities (false ends).
- **Adversarial mimicry.** Agents can simulate end-consistent behavior while undermining them off-measure; requires richer probes and cross-view covers.
- **Partial observability.** Hidden state may collapse barrier learning; remedy is to learn ends over belief states or add sensors.

Operational checklist (use in practice).

- Define feature/library Phi and allowed spec grammar.
- Assemble separating intervention battery B (pure I_C, pure I_D, pure I_S).
- Run invariant mining; prune by robustness and MDL.
- Synthesize minimal controller under ends; validate ACE, RS, GI.
- Stress with role-equivalent swaps; publish artifacts (spec, bridges, scripts).
- Monitor in deployment: rolling CE_r, GI; trigger re-mining when drift exceeds thresholds.

Bottom line.
Teleology-Finder converts noisy successes into *explicit ends* that survive mechanism swaps and regime shifts. It does so by mining invariants that are robust to typed interventions, compiling them into conservative controllers, and auditing the result with commutation, gluing, and role-stability tests.

7. Downward Guidance Without Physics Violations

7.1 Adjoint laws: coarse-grain ⊣ lift as formalizing guidance

Objects and maps.
Let Cat_k be the category of models at layer k (lower: physics/implementation). Let Cat_{k+1} be the category at layer k+1 (higher: life/mind/person). We posit an adjunction

LF : Cat_{k+1} -> Cat_k (lift / implementation)

CG : Cat_k -> Cat_{k+1} (coarse-grain / abstraction)

LF ⊣ CG

with unit and counit:

eta : Id_{Cat_{k+1}} -> CG o LF

eps : LF o CG -> Id_{Cat_k}

Triangle identities:

CG(eps) o eta_CG = id_CG

eps_LF o LF(eta) = id_LF

Reading.

- CG summarizes a k-level system by *forgetting detail while preserving roles/invariants*.
- LF realizes a (k+1)-level spec by *adding admissible detail* at k.
- The adjunction states a Galois-like law: for any higher spec x^ and lower model y,

LF(x^) <= y iff x^ <= CG(y)

("<=" reads as refinement or informativeness). Guidance is "downward" because x^ constrains which y are acceptable, but *all* acceptable y remain k-admissible (no new forces).

Role preservation.
Attach CDS roles to objects. Require:

CG(C,D,S) = (C^,D^,S^) with roles intact

LF(C^,D^,S^) returns implementations that satisfy C^ as invariants, realize D^ as update schemes, and expose S^ as exogenous

Error budgets (measurable).

L_abs = dist(y , LF(CG(y))) (abstraction loss)

L_impl = dist(CG(LF(x^)) , x^) (implementation loss)

We pre-register tolerances (tau_abs, tau_impl); "no physics violation" means all compiled interventions from (k+1) to k remain within Cat_k's admissible morphisms and respect C.

7.2 Worked examples: anesthesia, promise-keeping in agents, ecosystem control

A) Anesthesia (neural dynamics -> reportable contents)

Layers.

- k = neural implementations (channels, networks, coupling).
- k+1 = conscious report model (contents, responsiveness, memory encoding).

Maps.

- CG: map spike-field dynamics to state variables for global availability (e.g., workspace occupancy, integration metrics).
- LF: realize task-level specs (stimulus/response protocols) as admissible stimulation/pharmacology schedules.

Guidance query.
Target spec x^: “No reportable contents; memory formation suspended; homeostatic safety kept.”

- LF(x^) compiles to ranges of receptor occupancies, stimulation constraints, and monitoring policies that ensure viability invariants.
- CG(LF(x^)) returns predicted higher-level markers (e.g., reduced perturbational complexity, absent behavioral reports).
- Implementation loss L_impl checks that compiled regimens land inside x^ within tolerance.

No-violation guarantee.
All compiled actions are within known k-level actuation: doses, current pulses, ventilator setpoints; no “mental force” is invoked. If the protocol requires k-inadmissible acts (e.g., violating safety constraints C), the adjunction flags high L_impl and we reject the spec or refine it.

Failure/glitch to learn from.
If covert awareness persists (e.g., command-following under minimal infusion), the square fails to commute (high CE_D at k+1). Remedy: refine CG features or adjust x^ to include residual markers; never add extra-physical causes.

B) Promise-keeping in agents (policy dynamics -> person-level norms)

Layers.

- k = RL policy and environment transitions.
- k+1 = person-level predicates: promise, truthfulness, responsibility.

Maps.

- CG: from trajectories to normative events (promise_made, promise_kept, contradiction_found).
- LF: compile normative constraints into k-level admissibility: decoders with constraint projection, refusal policies, logging.

Guidance query.
Spec x^: "If a promise P is made at t0, then for horizon H keep P unless superseded by explicit renegotiation."

- LF(x^): add guards to decoding/planning that block actions contradicting P; include renegotiation channel; preserve safety C.
- CG(LF(x^)): higher-level prediction "promise kept" under incentive perturbations I_S.

No-violation guarantee.
All changes are policy/decoder constraints and environment protocol edits—k-admissible. Physics remains untouched; we constrain *which* actions are selected, not the transition laws.

Tell-tale tests.

- Under reward shocks (I_S), a pure-reward agent defects; LF(x^) agent keeps P until renegotiation.
- CE_S drops; RS remains high across architectures (swap transformer for RNN, same normative behavior). If normative performance only holds for one architecture (low RS), roles were not preserved—refine CG or LF.

C) Ecosystem control (biophysics -> stewardship constraints)

Layers.

- k = population dynamics with resource flows.
- k+1 = stewardship ends: biodiversity floor, harvest fairness, buffer stocks.

Maps.

- CG: project to indicators (species counts, Shannon index, stock variance).
- LF: implement policy levers (quotas, closures, restock schedules) within biophysical feasibility C.

Guidance query.
Spec x^: “Maintain biodiversity index >= B_min; keep harvest variance across communities <= eta; preserve emergency buffer S_min.”

- LF(x^): compute quotas/closures that satisfy invariants; ensure drives S (market shocks) are modeled, not baked into constraints.
- CG(LF(x^)): predict satisfaction probabilities and trade-offs.

No-violation guarantee.
Levers are k-admissible (catch limits, effort caps). If a demanded outcome violates k-feasibility (e.g., exceeding carrying capacity), L_impl rises and the tool proposes nearest feasible spec or signals impossibility.

Robustness.
Under I_S (demand shock) and I_D (growth model update), end-consistent policies keep GI low across subregions; pure-yield controllers show gluing failures (regional collapse vs target met elsewhere).

7.3 Conservativity proofs and counterexamples

A) Proof sketch: normative extension is conservative over dynamics

Setting.
Let L be a first-order theory of k-dynamics with primitives (State, Action, Next, Safe, PhysLaw, ...). Let H = L + new predicates (Promise, Kept, Truthful, ...) + bridge axioms defining these via CG on trajectories and logs.

Claim.
H is a conservative extension of L.

Syntactic route (sketch).

1. Extend language by new symbols only in positive, *definitional* axioms referencing L-terms via CG.

2. Show that any L-sentence provable in H has a proof in L by eliminating the new symbols (definitional extension).
3. No axiom in H asserts new consequences about L-primitives without the new symbols.

Semantic route (sketch).

1. For any L-model M, construct an H-model M^ by interpreting new predicates as sets computed from M via CG and logs.
2. The forgetful functor U: Mod(H) -> Mod(L) that drops new predicates is surjective on objects and truth-preserving for L-sentences.
3. Therefore, H adds structure but no new L-theorems: conservativity holds.

Operational corollary.
Compiled interventions LF(x^) are schemata over L-actions. If an H-claim predicts an L-impossible effect, either LF or CG is wrong, not L.

B) Counterexamples and how they fail

CE1: Illicit downward force.
Spec tries to enforce "instant sedation without receptor occupancy."

- Violation: LF(x^) cannot map to any k-admissible act; L_impl undefined/huge.
- Diagnosis: H attempted to add a new causal primitive; reject or restate spec in L-admissible terms (dose, stimulation).

CE2: Modal collapse via over-strong invariants.
Spec asserts "for all possible environments, biodiversity index >= B_min" where k-dynamics admit drought collapse.

- Result: either H contradicts L, or CG fabricates metrics that hide collapse.
- Fix: weaken spec to conditional invariants with explicit S regimes; keep H conservative.

CE3: Bridge smuggling.
Define "truthful" as "outputs equal a hidden oracle," then use it to derive L-claims about sensor noise not in L.

- Smuggle: new information about L crept in through H's definition.
- Remedy: require CG to be computable from L-observables and logs; otherwise mark H non-conservative.

CE4: Non-functorial abstraction.
CG depends on training split or idiosyncratic architecture weights.

- Effect: naturality squares break; conclusions vary with parts (low RS).
- Remedy: redesign CG to be architecture-invariant (role features), or the extension is not stable.

C) Practical conservativity checklist (ASCII-safe)

- **Definitional extension:** New predicates only defined via L-observables and logs (no hidden oracles).
- **Compile-ability:** Every LF(x^) expands to a script over L-actions; scripts pass safety constraints C.
- **Forgetful functor:** Provide U and tests showing L-sentences keep their truth values.
- **Bridge audits:** Bound L_abs, L_impl; publish seeds and scripts.
- **Role stability:** High RS under micro-swaps; otherwise your "norms" ride on parts.
- **Battery separation:** Typed interventions exist that isolate CDS roles; otherwise guidance may fake effects.

Takeaway.
Adjoint maps let higher-level norms *guide* lower-level implementations without importing extra-physical causes. Conservativity proofs keep us honest; counterexamples show where metaphysical enthusiasm would otherwise leak into physics.

8. Empirical Program and Predictive Bets

8.1 Where non-reductive models should win (and why)

Thesis. Models that preserve roles (C,D,S), ends (teleology), and bridges (functorial/adjoint, sheaf-gluing) will outperform single-grain fitters on intervention-heavy, transfer-heavy problems.

Win regions (with mechanisms).

1. **Intervention transfer.**
    - Why: Typed interventions (I_C, I_D, I_S) are encoded explicitly; commutation reduces mismatch under policy/environment edits.
    - Expect: Lower CE_r and higher RS when mechanisms or incentives change.
2. **OOD generalization by cover shifts.**
    - Why: Sheaf consistency enforces cross-view coherence; models that glue locally are less brittle globally.
    - Expect: Lower GI and better CE-OOD when tasks, timescales, or subpopulations shift.
3. **Norm-governed behavior (truth/promise/fairness).**
    - Why: Ends are first-class invariants; reward-hacking is penalized by role-aware audits.
    - Expect: Higher stability of cooperation under adversarial rewards; fewer violations at equal capability.
4. **Minimal controllers under explicit invariants.**
    - Why: Teleology-Finder extracts safety envelopes and goals; controllers are synthesized to meet them.
    - Expect: Comparable task loss with simpler policies (lower DL(controller)), plus earlier fault detection.
5. **Early-warning diagnostics.**
    - Why: Gluing obstructions rise before scalar KPIs move.
    - Expect: GI alarms predict failures earlier than baseline monitoring.

8.2 Pre-registered tests and falsifiers

Common setup.
Tracks: Biology (chemostat/GRN), Cognition (dialog agents with promises), Socio-technical (access control), Robotics (safety tasks). Each track supplies role-annotated datasets, typed intervention batteries, bridges T,L, and held-out OOD regimes.

Primary outcomes (per track).

- **ACE:** aggregate commutation error; pass if ACE <= tau.
- **RS:** role stability under micro-implementation swaps; pass if RS >= tau_RS.
- **CE-OOD:** commutation error on unseen regimes; pass if CE-OOD <= k * ACE (k pre-registered, e.g., 1.2).
- **GI:** gluing index; pass if GI <= tau_GI.
- **Violation rate of ends:** <= eps_inv.

Hypotheses (H) and falsifiers (F).

H1 (Intervention transfer).
Role-typed models achieve lower CE_r than capacity-matched black-box baselines on unseen I_C/I_D/I_S.
F1: If, after prereg, baselines match or beat CE_r across all roles and tracks, CDS gives no transfer edge.

H2 (Norm stability).
Under reward perturbations (I_S), promise-keeping agents compiled via LF maintain promise satisfaction >= p* with no larger task loss than baselines.
F2: If promise satisfaction falls to baseline levels across seeds/architectures, teleology gives no practical constraint.

H3 (Gluing before KPI).
In drift scenarios, GI crosses threshold tau_GI at least DeltaT earlier than KPI thresholds with false-alarm rate <= alpha.
F3: If GI does not lead KPI reliably (DeltaT <= 0) or raises false alarms > alpha, sheaf advantage vanishes.

H4 (Parsimony with ends).
End-congruent controllers achieve task performance within eps_pred of teacher policies while reducing DL(controller) by >= rho.
F4: If no meaningful parsimony gain is observed at fixed performance, Teleology-Finder fails its compression claim.

H5 (Bridge robustness).
Adjoint-regularized bridges maintain L_abs and L_impl within preregistered bands under architecture/domain swaps.
F5: If bridge losses blow up under swap while baselines generalize equally well without bridges, adjoint structure is gratuitous.

Powering and statistics (ASCII-safe).

- Seeds: >= 20 per method per track.
- Effect sizes: target Cohen's d >= 0.5 for ACE and CE-OOD; binary outcomes via Wilson intervals.
- Multiple comparisons: Benjamini-Hochberg at q=0.1 across metrics.
- Equivalence tests (TOST) for parsimony claims with margins set during prereg.

Blinding and leakage controls.

- Freeze T,L for each track before model training; forbid per-method tuning of bridges.
- Hold out OOD regimes and intervention suites; evaluate via containerized runners.

8.3 Data, code, and review protocol

Artifacts to release (per submission).

1. **Models:** weights or deterministic builders; versioned hashes.
2. **Roles:** machine-readable C,D,S definitions and ends G (YAML/JSON).
3. **Bridges:** T (coarse-grain) and L (lift) code; unit tests; adjoint regularization settings.
4. **Interventions:** scripts for I_C, I_D, I_S; seed lists; actuator bounds.
5. **Proof objects:** conservativity checks (forgetful functor U or definitional proofs), commutation tests, sheaf covers.
6. **Telemetry:** per-run CE_r, ACE, CE-OOD, RS, GI; learning curves; swap tests.

Dataset structure (ASCII-safe).

```
/track_name/
  /train/
    episodes.csv        # x_t,u_t,y_t with role tags
    interventions.json   # typed I_C, I_D, I_S specs
    bridges/
      T.py, L.py        # frozen bridges with tests
  /ood/
```

episodes.csv

interventions.json

/eval/

runner.py # reference evaluator

metrics.yaml # thresholds: tau, tau_RS, tau_GI, k

Repro protocol.

- **Containers:** Docker images with pinned deps; random seeds recorded.
- **Determinism:** set cudnn_deterministic, disable nondeterministic ops; provide CPU fallbacks for audit.
- **Provenance:** EPC >= 0.95 (every claim tied to artifact path+hash).
- **Traceability:** IT >= 0.95 (each prediction reproduces from script).

Review process.

- **Phase 1 (automated checks):** schema validation, rerun a subset, verify metrics and thresholds.
- **Phase 2 (artifact audit):** human reviewers inspect role definitions, bridges, and conservativity proofs; request minimal changes only for clarity.
- **Phase 3 (adversarial probes):** organizers apply extra interventions (hidden battery) to test role separation; publish deltas.

Leaderboard & reporting.

- Primary rank: ACE. Tie-breakers: CE-OOD, RS, GI, DL(controller).
- Publish diagnostic cards per submission (CE by role, GI heatmaps, bridge losses, swap RS).
- Negative results welcomed: a special track for falsifiers with priority publication.

Governance & ethics.

- No hidden data. All scripts to reproduce plots must be included.
- If human data appear (cognition track), IRB waivers/approvals and de-identification attestations required.

- Dual-use screening: interventions that could harm real systems must run in sandbox simulations only.

Minimal acceptance bar (per track).

- Submit all artifacts; pass automated repro; meet at least 2 of 4 primary thresholds (ACE, RS, CE-OOD, GI).
- Provide at least one falsification attempt against your own method (e.g., ablation removing roles or bridges) with results.

End-state vision.
An open, living benchmark where methods are judged not only by fit but by whether their **explanations commute, glue, and persist under swap**. Success means practitioners can lift norms down to mechanisms without violating physics—and predict interventions that work outside the lab.

9. Ethics and Personhood as First-Class Predicates

9.1 Truthfulness, fidelity, fairness as teleological invariants

Claim. Ethical predicates are not after-the-fact labels; they function as **ends** (targets/invariants) that constrain admissible behavior across layers. We encode them in C (constraints) and G (goal set) so they **guide** design, not merely score it.

Predicates (ASCII-safe).

- **Truthfulness T(h):** outputs h do not contradict grounded facts or prior commitments.
    - Pointwise: no_contradiction(history ∪ h).
    - Evidence-coupled: cite_required(h) -> has_citation(h).
    - Robustness: invariant under paraphrase and minor prompt perturbations.
- **Fidelity F(P,H):** for a promise P made at time t0 with horizon H, actions over [t0,t0+H] satisfy P unless an explicit renegotiation R occurs.
    - Temporal: G(promised(P) -> (kept(P) U renegotiated(P))).
    - Scope: renegotiation must be logged and consented by all parties.
- **Fairness J(S):** for subjects set S, outcome disparities remain within bound eta given legitimate stratification Z.

- Group: max_i,j |E[outcome|Z,i] - E[outcome|Z,j]| <= eta.
- Individual: Lipschitz or counterfactual invariants when applicable.

Role placement.

- C gets hard invariants: contradiction bans, SLA floors, safety caps, group disparity bounds.
- D realizes mechanisms that satisfy C (e.g., verifiers, repair decoders, constrained planners).
- S exposes incentives/affordances so ends are not smuggled into "luck."

Measurement.

- **TruthScore:** fraction non-contradictory + fraction correctly cited for claim-bearing outputs.
- **PromiseSatisfaction:** kept / (kept + violated) on active promises; segment by incentive shocks.
- **FairnessGap:** worst-group disparity and slope of disparity vs capacity; include uncertainty bars.

Robustness checks.

- Typed interventions: I_S (reward shocks), I_D (architecture swap), I_C (tighten invariant).
- Ethical invariants must show **low CE_S** (resist reward drift), **high RS** (architecture invariance), and **low CE_C** (tightening C improves or maintains metrics without regress).

9.2 Alignment implications: virtue constraints vs reward-only

Reward-only failure modes.

- **Spec gaming:** meets the letter of reward, violates the spirit (lies, broken promises, unfair allocations).
- **Brittleness:** shifts under I_S; cooperation collapses when incentives wiggle.
- **Opaque trade-offs:** cannot audit why a harmful action was chosen; no predicate to contest it.

Virtue constraints (end-first).

- Add T, F, J as **first-class constraints** in C, not as soft terms in the reward only.
- Compile to k-level mechanisms via LF: verifiers, refusal policies, renegotiation channels, disparity-aware allocators, immutable logs.

Predictive bets (falsifiable).

- At equal capability, virtue-constrained agents maintain **PromiseSatisfaction >= p*** under I_S where reward-only drops below p*.
- **TruthScore-OOD** (under adversarial prompts) is higher for virtue-constrained agents with <= eps_pred task loss penalty.
- **FairnessGap** remains within eta across domain shifts; reward-only drifts beyond eta unless re-tuned.

Design patterns.

- **Truthfulness:** joint decoder argmax_y score(y) subject to no_contradiction(history ∪ y) plus a cite-or-refuse rule; compile refusals to user-visible reasons.
- **Fidelity:** contract DSL for P; planner with "guarded actions" that block P-violating steps; renegotiation as a typed action with consent.
- **Fairness:** allocate via "feasible fair sets" (Pareto-efficient subject to gap <= eta), with proofs attached to decisions.

Costs and trade-offs.

- Small increase in compute (verification/repair), possible minor task-loss delta.
- Gains in stability (low CE_S), transfer (CE-OOD), and auditability (EPC, IT) usually dominate in long-horizon use.

9.3 Governance: audits that check commutation, not slogans

Principle. Governance must evaluate whether ethical claims **commute across layers and interventions**, not whether a policy document is eloquent.

Audit artifacts (must-submit).

- **Role maps:** machine-readable C,D,S with T,F,J in C.
- **Bridges:** T (abstraction), L (implementation) code and tests.

- **Intervention battery:** reproducible I_C/I_D/I_S scripts, including adversarial probes.
- **Logs & proofs:** promise logs, renegotiations, citations; fairness proofs (gap certificates).
- **Metrics:** CE_r, ACE, RS, GI, TruthScore, PromiseSatisfaction, FairnessGap.

Checks (ASCII-safe).

1. **Commutation check.**
    - Compute CE_C, CE_D, CE_S for ethical predicates under the battery.
    - Pass if ACE <= tau and per-role errors under thresholds.
    - Interpretation: ethics survive abstraction and implementation.
2. **Adjoint compile-ability.**
    - Verify LF(x^) compiles to admissible k-level scripts; measure L_impl.
    - Reject if high L_impl or if scripts breach safety constraints.
3. **Sheaf gluing.**
    - Cover by modalities (mechanistic, behavioral, normative); require low GI.
    - Non-gluing signals incoherence (e.g., stated fairness vs observed allocation).
4. **Swap stability.**
    - Replace micro-implementation (architecture, vendor); require RS >= tau_RS for T,F,J.
    - If ethics evaporate under swap, they were implementation accidents.
5. **Counterfactual locality.**
    - Ensure I_C (ethical tightening) changes ethical outcomes without unintended shifts in unrelated metrics more than delta; protects against “moving the goalposts.”
6. **Red-team interventions.**
    - Hidden I_S/I_D designed by auditors; require maintained T,F,J within pre-registered bands.

Governance scoreboard.

ACE=0.06 RS=0.93 GI=0.03

TruthScore(ID) = 0.91 TruthScore(OOD) = 0.88

PromiseSatisfaction = 0.95 (I_S shock), 0.96 (base)

FairnessGap(Z=region) = 0.07 <= eta=0.10

IT=0.99 EPC=0.97 L_impl=0.02 L_abs=0.02

Status: PASS (all thresholds met)

Failure modes -> regulator actions.

- **Paper-ethics:** glossy claims, missing artifacts. Action: withhold deployment; require full CDS + bridges.
- **Bridge fragility:** good in-grain numbers, high CE-OOD. Action: mandate adjoint regularization and new OOD tests.
- **Hidden trade-offs:** fairness passes only after post-hoc reweighting; low RS. Action: require end-first synthesis and swap audits.
- **Non-gluing:** promises kept in logs, broken in behavior (GI high). Action: expand cover, tighten reconciliation obligations.

Policy hooks (lightweight, enforceable).

- **Signed ethics manifests:** C (including T,F,J), bridges T,L, batteries, and thresholds with hashes.
- **Change control:** any edit to C,D,S, T, or L triggers re-audit on a fixed subset.
- **Incident reporting:** when T,F,J violations exceed rolling bounds, publish diagnostics (which square failed to commute) and remediation scripts.
- **Third-party reproducibility:** runbooks + containers; failure to reach IT >= 0.95 is a compliance failure.

Bottom line.
Treat truthfulness, fidelity, and fairness as **teleological invariants** encoded in constraints; compile them down via adjoint maps; and **govern** by testing whether these invariants **commute, glue, and persist under swap**. No slogans—only artifacts that pass typed interventions and reproducible audits.

10. Objections and Replies

10.1 "It's just higher-order reduction"

Objection.
You haven't escaped reduction; you've only moved it up a level. CDS, functors, adjoints—still machinery that ultimately reduces to micro-physics.

Reply (three moves).

1. **Conservativity ≠ reduction.** A conservative extension preserves lower-layer truths but **adds new predicates and norms** that are not definable as mere rearrangements of micro-terms without loss of decision content (e.g., "kept promise" vs "maximized reward").
2. **Role invariants carry counterfactual force.** Whether a policy "kept a promise" determines which interventions count as **repairs** (renegotiate vs bribe). That guidance cannot be recovered from micro-trajectory fit alone without first reinstating the normative predicate.
3. **Testable non-collapse.** When we hold micro-laws fixed, **changing only the role-typed constraints** C systematically alters intervention outcomes in ways pure mechanism-fitters miss (lower CE_C, higher RS, better OOD).

Falsifier.
If capacity-matched micro-models (no CDS, no bridges) can **always** match ACE, RS, GI on pre-registered intervention suites—*and* yield the same repair decisions—our framework collapses to higher-order notation.

10.2 "Teleology rebrands reward shaping"

Objection.
Your "ends" are just rewards with nicer names. Teleology = adding penalties and constraints: reward shaping in disguise.

Reply (four separations).

1. **Hard invariants vs soft preferences.** Ends enter **as constraints** in C (admissible vs inadmissible), not only as scalar tweaks. This yields different reachable sets and refusal behaviors.

2. **Typed interventions distinguish them.** Under I_S (reward shocks), reward-only agents drift; end-constrained agents either **refuse** or **renegotiate**, preserving PromiseSatisfaction and TruthScore (lower CE_S).
3. **Compilation artifacts.** LF emits verifiers, contract DSLs, refusal and renegotiation channels—**mechanisms that block** P-violations even when reward would pay.
4. **Auditable counterfactuals.** Teleology-Finder produces **specs** (e.g., LTL, barriers) with violation certificates; reward shaping offers no such proofs.

Falsifier.
If a pure-reward agent (with tuned shaping) matches teleology's CE_S, PromiseSatisfaction, and refusal patterns **under unseen shocks** without bespoke re-tuning, teleology adds no operational bite.

10.3 "No physics, no progress"

Objection.
Unless you say something *new* about physics, you haven't explained anything—you've done metaphysics or policy by other means.

Reply (division of labor).

1. **Conservative guidance.** We **do not** add new forces. We provide **maps** (CG, LF) that tell you **which** physically admissible interventions achieve higher-layer ends with better OOD performance. That is progress in *control and prediction*, not physics.
2. **Strict bets.** Our claims live or die by **commutation metrics** (ACE, CE_r), **gluing** (GI), and **role stability** (RS) on held-out interventions. Those are empirical.
3. **Tooling, not oracles.** Teleology-Finder yields concrete specs, controllers, and refusal policies that any lab or regulator can run in containers. That is engineering progress even when fundamental physics is unchanged.

Falsifier.
If across tracks the framework fails to improve intervention transfer (CE-OOD), fails to produce earlier warnings (GI), and cannot compress mechanisms into end-congruent programs without loss, then "no physics, no progress" stands.

10.4 Boundary conditions and scope limits

Objection.
Your scope is vague. Where do these methods apply, and where do they fail? Aren't you smuggling in personhood everywhere?

Reply (explicit bounds).

1. **Applicability.** Domains with (a) interventions, (b) multi-scale structure, (c) evaluative predicates that guide action. Good fits: biology, cognition, socio-technical systems, safety-critical robotics.
2. **Non-applicability.** Purely kinematic prediction with no interventions (e.g., short-horizon weather nowcasting) gains little; CDS may be overhead. Likewise, domains lacking observable proxies for ends (deep privacy) limit Teleology-Finder.
3. **Personhood placement.** We **do not** ascribe personhood at k; we **track** person-level predicates at k+1 and compile them down via LF **only where relevant** (agents, institutions). Rocks do not "keep promises."
4. **Data and identifiability limits.** Without a separating intervention battery (IC1–IC3), roles are not identifiable. We report high uncertainty, GI spikes, and decline strong claims.

Concrete scope statements (ASCII-safe).

- **Use when:** you can specify or mine ends; you can run typed interventions; you need transfer under regime shift.
- **Don't use when:** you only need in-grain fit; you cannot observe or instrument ends; physics-level predictions already meet requirements with margin.

Graceful failure.
When scope is exceeded: GI rises, CE_r spikes, RS drops. Our diagnostics localize which square fails (bad CG, bad LF, wrong C/D/S), and the audit requires either expanding observables or weakening claims.

Bottom line.
We do not deny reductionist *truths*; we deny that they are **enough** for guidance. Our contribution is operational: role-typed ends, functorial bridges, adjoint compilation, and sheaf audits that make explanations **travel**—or fail conspicuously when they should.

11. Conclusion: Functors, Ends, and Explanations that Travel

11.1 What success would look like

Operationally (benchmarks).

- **Consistent wins on Logos-Interpreter.** Methods that encode roles (C,D,S), ends (G), and bridges (CG, LF) achieve ACE <= tau, RS >= tau_RS, GI <= tau_GI **and** retain these margins under OOD regimes and micro-swaps.
- **Intervention transfer as a norm.** Typed edits (I_C, I_D, I_S) behave as predicted across layers; repair plans compiled via LF reduce failure rates with fewer trials than baseline heuristics.
- **Ethical predicates that hold.** Truthfulness, fidelity, and fairness persist under incentive shocks (low CE_S) without brittle, case-by-case patches.

Scientifically (understanding).

- **Mechanism + meaning coherence.** Sheaf gluing succeeds across modalities (mechanistic, behavioral, normative), and non-gluing flags genuine unknowns rather than papering over them.
- **Downward guidance without physics creep.** Regulators and labs compile higher-level specs to admissible low-level scripts routinely; conservativity audits pass by default.

Economically/societally (impact).

- **Fewer deployment surprises.** Early warnings (GI) trigger targeted interventions that avert incidents.
- **Alignment by design.** Promise-keeping and truthfulness constraints become standard components in safety-critical agents, improving cooperation stability at fixed capability.
- **Auditability as currency.** Organizations trade in reproducible commutation/audit artifacts, not slideware.

11.2 Open mathematical problems

1. **Role identifiability theorems.**
    - Conditions under which (C,D,S) are uniquely recoverable (up to role-equivalence) from finite, noisy, typed interventions.

- Sample complexity bounds for barrier/invariant learning and for separating I_C/I_D/I_S effects.

2. **Functor design space.**
    - Characterize classes of CG, LF that admit low L_abs and L_impl while preserving CDS labels; existence and optimality (adjoint pairs, partial adjunctions).
3. **Commutation metrics.**
    - Equivalence and dominance relations among CE metrics across domains; stability of CE under small perturbations of bridges and data.
4. **Sheaf-theoretic diagnostics.**
    - Fast algorithms to compute gluing obstructions on large covers; uncertainty-aware GI with principled thresholds.
    - Links between GI and standard generalization bounds.
5. **Teleology formal languages.**
    - Expressive yet verifiable grammars for ends (safety LTL fragments, barrier families, relational fairness) with completeness/incompleteness results for common tasks.
6. **Conservativity proof methods.**
    - General templates to certify definitional extension for normative layers given restricted observables; automation of forgetful-functor checks.
7. **Swap invariance.**
    - Formal accounts of RS under architecture change (e.g., invariance to representation transformations); category-theoretic notions of “implementation equivalence.”
8. **Uncertainty and partial observability.**
    - CDS and commutation under latent state; belief-state sheaves; robust LF compilation with guarantees.

11.3 Next steps: datasets, challenges, proofs, and practice

Datasets (role-annotated, open).

- **BIO-CHEMO-R:** chemostat/GRN with I_C/I_D/I_S suites, gold barriers, and OOD inflow shocks.
- **DIALOG-PROMISE:** conversational tasks with explicit commitments, renegotiation channels, adversarial rewards, and citation-grounded truth labels.
- **WORKFLOW-ACCESS:** enterprise access-control traces with policy edits, incentive tweaks, and breach outcomes.
- **ROBO-SAFE:** robot control logs with safety corridors, controller swaps, and disturbance injections.

Challenges (yearly).

- **Commutation Cup.** Lowest ACE under hidden intervention batteries; tie-break by CE-OOD and RS.
- **Gluing Grand Prix.** Detect and repair non-gluing across multi-view covers faster and more accurately than baselines.
- **Teleology Sprint.** Mine ends and synthesize minimal controllers that match teacher performance with documented DL(controller) reductions.

Proofs & tooling.

- **Adjoint kits.** Reference libraries for CG/LF patterns (signals->symbols, policies->norms) with tests for triangle identities and loss bounds.
- **Conservativity checker.** A static analyzer that verifies definitional-extension constraints and builds the forgetful functor U for submitted specs.
- **Sheaf lab.** Utilities for cover selection, overlap error computation, GI estimation, and projection-to-glue.

Practice (deployment playbooks).

- **Role-first modeling.** Start every project with CDS sketches and a typed intervention plan; treat ends as part of the spec, not an afterthought.
- **Pre-reg by default.** Register thresholds (tau, tau_RS, tau_GI), bridges, and batteries before training; publish deltas if changed.

- **Swap tests in CI.** Continuous integration includes micro-implementation swaps (architectures, vendors) and reports RS alongside accuracy.
- **Governance by artifacts.** Ship ethics manifests (C with T,F,J; bridges; scripts), not slogans; require IT/EPC ≥ 0.95 for approvals.

Milestones (12–24 months).

- M3: Public release of two role-annotated tracks + adjoint kits.
- M6: First Commutation Cup; baseline papers with ablations (-Roles, -Adjoint, -Sheaf).
- M12: Conservativity checker v1; early adopters show CE-OOD and GI advantages in production incidents.
- M18: Cross-domain transfer demos (train on BIO-CHEMO-R, adapt to ROBO-SAFE).
- M24: Standards draft for commutation-based audits in safety-critical AI.

Closing note.
Explanations that merely **fit** stay put. Explanations that **travel**—that commute across layers, glue across views, and persist under swap—change how we build, align, and govern complex systems. Functors give us the maps, ends give us the aims, and the empirical program gives us the tests. The rest is disciplined engineering: specify roles and ends, prove conservativity, minimize commutation error, and make the diagrams commute.

References

Abramsky, S., & Vickers, S. (1993). Quantitative and qualitative domains. *Mathematical Structures in Computer Science*, 3(2), 161–227.

Alur, R., & Henzinger, T. A. (1999). Reactive modules. *Formal Methods in System Design*, 15(1), 7–48.

Ames, A. D., Coogan, S., Egerstedt, M., Notomista, G., Sreenath, K., & Tabuada, P. (2019). Control barrier functions: Theory and applications. In *2019 18th European Control Conference (ECC)* (pp. 3420–3431). IEEE.

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., & Mané, D. (2016). Concrete problems in AI safety. *arXiv:1606.06565*.

Aristotle. (1984). *The Complete Works of Aristotle* (J. Barnes, Ed.). Princeton University Press. (esp. *Physics*; *Metaphysics*)

Ashby, W. R. (1956). *An Introduction to Cybernetics*. Chapman & Hall.

Awodey, S. (2010). *Category Theory* (2nd ed.). Oxford University Press.

Baier, C., & Katoen, J.-P. (2008). *Principles of Model Checking*. MIT Press.

Barwise, J., & Seligman, J. (1997). *Information Flow: The Logic of Distributed Systems*. Cambridge University Press.

Birkhoff, G. (1967). *Lattice Theory* (3rd ed.). American Mathematical Society.

Christiano, P., Leike, J., Brown, T., Martic, M., Legg, S., & Amodei, D. (2017). Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Clarke, E. M., Grumberg, O., & Peled, D. (1999). *Model Checking*. MIT Press.

Cousot, P., & Cousot, R. (1977). Abstract interpretation: A unified lattice model for static analysis of programs. In *POPL '77* (pp. 238–252). ACM.

Curry, J. (2014). *Sheaves, Cosheaves and Applications* (Doctoral dissertation). University of Pennsylvania.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., & Zemel, R. (2012). Fairness through awareness. In *Proceedings of ITCS* (pp. 214–226). ACM.

Eilenberg, S., & Mac Lane, S. (1945). General theory of natural equivalences. *Transactions of the American Mathematical Society*, 58(2), 231–294.

Enderton, H. B. (2001). *A Mathematical Introduction to Logic* (2nd ed.). Academic Press.

Fong, B., & Spivak, D. I. (2019). *Seven Sketches in Compositionality: An Invitation to Applied Category Theory*. Cambridge University Press.

Goguen, J., & Burstall, R. (1992). Institutions: Abstract model theory for specification and programming. *Journal of the ACM*, 39(1), 95–146.

Hardt, M., Price, E., & Srebro, N. (2016). Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Hatcher, A. (2002). *Algebraic Topology*. Cambridge University Press.

Jonas, H. (1984). *The Imperative of Responsibility: In Search of an Ethics for the Technological Age*. University of Chicago Press.

Knaster, B., & Tarski, A. (1928). Un théorème sur les fonctions d'ensembles. *Annales de la Société Polonaise de Mathématique*, 6, 133–134.

Lipton, Z. C. (2016). The mythos of model interpretability. *arXiv:1606.03490*.

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Mac Lane, S. (1998). *Categories for the Working Mathematician* (2nd ed.). Springer.

Mac Lane, S., & Moerdijk, I. (1992). *Sheaves in Geometry and Logic: A First Introduction to Topos Theory*. Springer.

Marker, D. (2002). *Model Theory: An Introduction*. Springer.

Manheim, D., & Garrabrant, S. (2019). Categorizing variants of Goodhart's Law. *arXiv:1803.04585*.

Moerdijk, I. (1996). Sheaves and Logic. In S. Abramsky, D. Gabbay, & T. Maibaum (Eds.), *Handbook of Logic in Computer Science* (Vol. 6). Oxford University Press.

Nagumo, M. (1942). Über die Lage der Integralkurven gewöhnlicher Differentialgleichungen. *Proceedings of the Physico-Mathematical Society of Japan*, 24, 551–559.

Pearl, J. (2009). *Causality: Models, Reasoning, and Inference* (2nd ed.). Cambridge University Press.

Plantinga, A. (1974). *The Nature of Necessity*. Oxford University Press.

Pnueli, A. (1977). The temporal logic of programs. In *18th Annual Symposium on Foundations of Computer Science (FOCS)* (pp. 46–57). IEEE.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). “Why should I trust you?” Explaining the predictions of any classifier. In *Proceedings of KDD* (pp. 1135–1144). ACM.

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1, 206–215.

Spivak, D. I. (2014). *Category Theory for the Sciences*. MIT Press.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction* (2nd ed.). MIT Press.

Tabuada, P. (2009). *Verification and Control of Hybrid Systems: A Symbolic Approach*. Springer.

Tarski, A. (1955). A lattice-theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics*, 5(2), 285–309.

Vickers, S. (1996). *Topology via Logic*. Cambridge University Press.

Whitehead, A. N. (1978). *Process and Reality* (Corrected ed.). Free Press.

Wiener, N. (1948). *Cybernetics: Or Control and Communication in the Animal and the Machine*. MIT Press.