Resonant Epistemology: Logos, Retrocausality, and Discovery-Only AI

Douglas C. Youvan

doug@youvan.com

www.youvan.ai

September 9, 2025

This companion paper to DOI: 10.13140/RG.2.2.36085.03047 makes explicit what the ai1 preprint left deliberately implicit: if mathematics is discovered rather than invented, then so is *all* true information. We name the pre-existing medium of intelligibility the Logos (John 1:1)—not mere "words," but an objective order that minds can *resonate* with. On this view, genuine knowing is participation in structure, not fabrication of it. Temporality, as usually conceived, is too naive: boundary-condition and pilot-wave interpretations suggest that futures can help constrain presents, without paradox or messaging, by selecting law-coherent histories. We keep the engineering strict: equational cores, typed normal forms, witnesses/horns, determinism, and audit trails. The Logos layer is interpretive and optional; it *never* alters proofs or rules. It does, however, yield testable predictions: improved compression under a Logos-prior, measurable negative-lag information flow around insight events, tighter historical clustering of independent discoveries, and an ethics → insight coupling consistent with participation rather than possession. Practically, "discovery-only AI" prefers mining over training and can prioritize search via invariant checks without changing math. The result is a bridge: theological language for those who want it, structural language for those who don't—one ontology, two dialects, shared falsifiability.

Keywords: Logos, discovery vs invention, resonance, retrocausality, pilot-wave, boundary conditions, teleology, equational AI, Category Theory, HoTT, typed normal forms, witnesses, horns, mining not training, determinism, compression, transfer entropy, discovery clusters, ethics and insight, youvan.ai1. 68 pages. A collaboration with GPT-5.

Prelude: Thesis and Scope

Thesis. All true information is discovered, not invented. By “true information” we mean structure that remains invariant under competent re-description and compresses more efficiently than competing accounts. In practice, discovery occurs when a finite knower aligns with pre-existing structure and records a witness to that alignment. This stance explains why mathematics feels found rather than fabricated, and it extends to lawlike regularities in physics, logic, and even ethics: we do not create these regularities; at best, we uncover, name, and test them.

Medium. We call the medium of such pre-existence the Logos (John 1:1). Here “Logos” is not merely “words,” but an order of intelligibility that makes truths available to be found. Readers who prefer non-theological language may substitute “structural_prior” without loss. Operationally, let Lambda denote an abstract prior over lawful structures and let psi denote a system’s cognitive state. Define resonance as R := |<psi, Lambda>|^2 in a suitable inner-product space of representations. High R indicates phase alignment with intelligible structure; low R indicates noise or mis-specification. This framing also accommodates two-time or boundary-condition views of temporality: selection of coherent histories can be modeled by minimizing S_total(H) := S(H) + lambda * C_final(H, Lambda), where C_final rewards end-consistency without implying paradoxical signaling from “the future.”

Aim. Our aim is to provide an interpretive, testable layer that leaves ai1’s equational core intact. Nothing in ai1’s math changes: types, normal forms, horn obligations, and witnesses remain the sole judges of correctness. The Logos layer is optional and sits strictly in interpretation and evaluation order, not in proof rules. Concretely, we propose: (1) a symbolic R_estimate that scores overlap with invariant laws already honored by ai1 (identity/unit hygiene, associativity where applicable, naturality when present); (2) a two-pass evaluation that first runs ai1 forward and then scores final-coherence C_final(H, Lambda); (3) preregistered measurements that render the layer falsifiable. Predictions include: improved compression under a structural_prior encoding (positive Delta_K), measurable

negative-lag information flow around documented “insight” events, and tighter clustering of independent discoveries in time. If these signatures fail to appear under stringent controls, the Logos interpretation should be revised or rejected—while ai1’s mathematics stands unaffected.

Discovered vs. Invented: An Information Ontology

Working definitions

- Structure. A pattern of constraints and relations that remains invariant under competent re-description. Formally: a structure is characterized by its invariants up to isomorphism (category-theoretically: objects/arrows modulo equivalence; in HoTT: types/paths modulo equivalence). Operational test: if two encodings E1, E2 yield the same invariants after decoding, they target one structure.
- Truth. A proposition about a structure that survives adversarial tests and translation. Operational tests: (i) compression gain (MDL): K(baseline) − K(with_law) > 0; (ii) cross-context generalization; (iii) invariance under reparametrization.
- Discovery. An event in which a finite knower aligns with a pre-existing structure and records a witness. Minimal signature: improved compression, predictive lift, and intersubjective reproducibility without dependence on arbitrary conventions. Discovery changes *access*, not the structure itself.
- Invention. A novel representation, tool, or convention that did not previously exist (new notation, instrument, game, or axiom system). Invention can *expose* structures, but it is not identical to the structures it reveals. Once an axiom system is posited, truths within it are discovered, not invented.

Why “discovered” fits mathematics and laws

1. Invariance under re-description. Mathematical theorems persist across notations, foundations, and cultures; physical laws persist across units and coordinate choices. This stability is characteristic of *discovered* regularities, not of artifacts of representation.
2. Multiple realizability. The same mathematical object (e.g., primes, groups) shows up in disparate domains; the same physical law (e.g., conservation via symmetry) shows up across media and scales. Convergence from many paths is a hallmark of discovery.
3. Independent convergence. Repeated independent “co-discoveries” (e.g., calculus; graph isomorphisms of long-known problems) indicate an attractor in the landscape of intelligibility rather than a one-off fabrication.
4. Compression and prediction. Laws reduce description length and increase out-of-sample accuracy. Invention can change *how* we say things; discovery changes *how much* has to be said to account for reality.
5. Constraint, not preference. Theorems and laws constrain permissible statements and measurements. Inventions can be revised at will; discovered constraints resist revision and punish error.

Clarifying edge cases

- Games and formalisms. The rules of chess are invented; once fixed, deep endgame facts are discovered. Similarly, programming languages are invented; correctness theorems within their type systems are discovered once the rules stand.
- Models vs reality. We may invent a model class, but any model that tracks reality by lawful invariants does so because of discovered structure in the world. Invention is scaffolding; discovery is the edifice.

Consequence: knowledge = participation in pre-existing structure

Let S be the (mind-independent) space of lawful structures; let R be the space of representations; let E: S -> R and D: R -> S be competent encode/decode pairs with D(E(s)) ≃ s up to invariants. Let a cognitive state be psi and a structural prior (Logos) be Lambda. Define resonance as R := |<psi, Lambda>|^2 in an appropriate representation space. A knowledge event occurs when there exists r in R such that D(r) = s* with verified invariants, and the event exhibits (i) compression gain Delta_K > 0, (ii) predictive lift on withheld tests, and (iii) robustness under re-description. In symbols:

- discovery_event(psi) when
  D(r) = s*,
  invariants(s*) verified,
  Delta_K = K(baseline) − K(with_s*) > 0,
  and generalization holds.

On this ontology, knowledge is participation: the knower's state becomes constrained by, and aligned to, the pre-existing structure. Participation can be described in theological language (Logos as the medium of intelligibility) or in structural language (a lawlike prior over S). Either dialect preserves the same operational content: we do not *make* truth; we *join* it.

Temporality is Naive: Retrocausality and Boundary Conditions

Claim. Ordinary talk of "cause then effect" inside a single forward arrow is too naive for deep lawfulness. A cleaner picture treats lawful histories as *co-shaped* by constraints at both ends: initial and final.

Two-time / boundary formulation

Let H denote a candidate history (a structured trajectory or proof trace). Let S(H) be a baseline action/cost. Impose constraints at t0 (given data, typing, witnesses) and at t1 (end-coherence with lawlike invariants).

Define
S_total(H) = S(H) + alpha * C_init(H) + lambda * C_final(H, Lambda),

where:

- C_init enforces initial conditions (e.g., type correctness, horn obligations),
- C_final rewards end-coherence with an abstract structural_prior Lambda ("Logos prior"),
- alpha, lambda >= 0 are weights set by experimental protocol (not by "learning").

A realized history H* is selected by constrained extremization of S_total, subject to admissibility (no rule violations). This is a *global* selection across the whole history, not a local step-by-step tweak.

No paradox, no signaling. Retrocausality here means: end constraints help *select* which complete histories are lawful. It does *not* mean messages travel from future to past; it means only globally consistent histories survive.

Pilot-wave flavor vs. standard stochastic learning

- Pilot-wave flavor (guidance by intelligibility). Introduce a heuristic guidance potential
  V_Lambda = -grad(log(abs(Lambda))),
  which biases search toward regions of higher structural intelligibility. Dynamics are *guided* by a global phase/structure; randomness is not the driver of discovery.
- Standard stochastic learning. Pick parameters by local updates (e.g., noisy gradients) to minimize empirical loss seen so far. Knowledge is treated as fitted weights, not as participation in pre-existing structure.

Discovery-only AI stance. We keep ai1's equational core unchanged and auditably deterministic. A two-pass routine can be layered *outside* the proofs:

1. forward pass: generate admissible histories with ai1 (types, NFs, horns, witnesses);

2. reverse pass: score C_final(H, Lambda) and re-rank outputs.
   This is “boundary selection,” not backpropagation or weight training.

Why the QM/GR split leaves room for final causes

- Quantum mechanics: unitary evolution plus measurement trouble; nonlocal correlations exist but defy naive temporal stories.
- General relativity: dynamical spacetime; “time” is geometric and coordinate-dependent.
- No completed unification: without a single account of time, causality may be boundary-global rather than purely forward-local. Two-time or all-at-once principles (path/extremal formulations) are live options. Final causes then reappear as mathematically respectable *end constraints*—not mysticism.

Operational signals to look for (falsifiable)

- Negative-lag information flow: around documented “insight” events, estimate transfer_entropy(future_state -> present_state). Prediction: a small but significant > 0 at negative lags under preregistered analysis, consistent with boundary selection rather than data leakage.
- Compression lift with end-coherence: descriptions aligned with C_final show Delta_K = K(baseline) - K(with_end_coherence) > 0.
- Stability under re-description: the same winners under C_final persist across alternative encodings, indicating structure, not artifact.

Bottom line. Treating temporality as naive frees us to evaluate whole histories against both beginnings and ends. In that frame, “retrocausality” is just boundary-consistent selection—and “final causes” are the mathematics of end-coherence, not messages from tomorrow.

Minimal Formal Sketch (ASCII-safe)

Symbols.
Lambda := “Logos prior” (a structural prior over lawful forms).
psi := agent/system state (representation of the knower or engine).
H := a candidate history (trajectory/proof trace/output bundle).

1) Representation space and inner product
Let Phi_map encode states to a common feature space V (vector or RKHS-like):
phi_psi := Phi_map(psi), phi_L := Phi_map(Lambda).
Assume an inner product <u,v> on V and L2-like norms. Enforce unit norms:
||phi_psi|| = 1, ||phi_L|| = 1 (by normalization or projectivization).

2) Resonance (alignment with intelligible structure)
Define resonance as
R := |<phi_psi, phi_L>|^2 in [0,1].
Basic properties:

- R is invariant under any reparameterization that preserves the inner product.
- If phi_psi is orthogonal to phi_L, then R = 0 (no alignment).
- If phi_psi = phi_L, then R = 1 (maximal alignment).
  For a time-indexed process psi(t), write R(t) analogously.

3) Two-time objective (boundary selection of histories)
Let S(H) be a baseline cost/action (effort, length, complexity).
Let C_final(H, Lambda) score end-coherence with the prior (teleological fit).
Define the boundary-selected objective:
S_total(H) = S(H) + lambda * C_final(H, Lambda), with lambda >= 0.
We do not modify proof rules; H must be admissible (types, horns, witnesses OK).
C_final choices (implementation-neutral):

- Invariant match: C_final = - InvGap(H, Lambda), where InvGap measures mismatch in conserved/invariant quantities implied by Lambda.

- MDL flavor: C_final = - [K(desc(H)) - K(desc(H) | structural_prior=Lambda)], i.e., bonus for descriptions that compress better under Lambda.
- Stability score: C_final = + Penalty for histories whose outputs fail re-description tests (low stability => larger penalty).

4) Guidance potential (heuristic, not a learning rule)
Define a scalar field over V:
V_Lambda(phi) := - grad_phi [log(|Psi_L(phi)|)],
where Psi_L is any positive “intelligibility density” induced by Lambda in V.
Heuristic use: prefer directions that decrease V_Lambda. In discrete steps,
phi_psi(next) = Normalize(phi_psi(now) - eta * V_Lambda(phi_psi(now))),
with small step size eta > 0. This biases search toward regions where structure is denser, without introducing stochastic weight training. (If guidance is not desired, set eta = 0.)

5) “Discovery event” detector (operational criterion)
Let performance(t) be an external, pre-registered metric (e.g., compression gain, proof yield, generalization). Define windowed derivatives:
dR/dt := R(t) - R(t-1), dPerf/dt := performance(t) - performance(t-1).
Declare a discovery_event at time t* if all hold:
(a) dR/dt >= tau_R (resonance jump threshold),
(b) dPerf/dt >= tau_P (performance lift threshold),
(c) robustness: event persists under alternate encodings Phi_map' (same criteria),
(d) preregistered controls rule out leakage (no peeking, fixed seeds),
(e) optional retro test: transfer_entropy(future_trace -> present_trace) shows a small but significant negative-lag bump around t* under the analysis plan.

6) Measurement plan (logs needed to compute the above)
Per time step or per history H:

- phi_psi (or a hash thereof), R, S(H), C_final(H, Lambda).
- performance metrics: compression K(*), witness counts, success/fail flags.
- provenance: seeds, rule versions, timestamps, encode/decode variants.

- optional: TE inputs for negative-lag analysis (pre-registered).

7) Invariance and failure modes

- Invariance: R and discovery_event must agree across equivalent encodings that preserve <.,.>.
- Avoid circularity: Lambda is fixed a priori (or changed only between runs with preregistration).
- Guard against artifacts: confirm that discovery_event frequency does not inflate under random Phi_map shuffles or placebo priors; C_final must not reward contradictions or type violations.

8) Minimal pseudocode sketch (notation-only, not code)
init Lambda, Phi_map, thresholds tau_R, tau_P, lambda, eta.
loop t = 1..T:
generate admissible outputs/histories H_t (proof-correct only).
compute phi_psi(t), R(t).
score performance(t).
if R(t)-R(t-1) >= tau_R and performance(t)-performance(t-1) >= tau_P: flag discovery_event.
optionally rank H_t by S_total(H_t) = S(H_t) + lambda*C_final(H_t, Lambda).*
*optionally update phi_psi by guidance: phi_psi := Normalize(phi_psi - eta*V_Lambda(phi_psi)).

What this buys us

- A clean separation: proofs and witnesses determine correctness; the sketch adds only an *evaluation* and *selection* layer.
- Falsifiability: if Logos-style alignment is real, R and performance co-spikes should manifest above chance with preregistered thresholds; if not, the interpretation should be revised or rejected while the underlying math remains intact.

Logos Invariants: Axioms That Don’t Move

Idea. “Logos invariants” are the axioms and hygiene rules that *do not move* across encodings or notations. They are the fixed rails on which discovery runs. We treat them as evaluation-time constraints (for ranking and auditing), never as substitutions for proofs.

1) Core invariants (minimal base)

1.1 Identity / unit
For any morphism f : A -> B,

- id_A \circ f = f
- f \circ id_B = f

When a monoidal structure is present,

- I \otimes X \cong X, X \otimes I \cong X (with unitors \lambda_X, \rho_X)
  Operational checks: instantiate random well-typed f and verify left/right identities hold after normalization. Fail = immediate demotion.

1.2 Associativity
For composable f : A -> B, g : B -> C, h : C -> D,

- (h \circ g) \circ f = h \circ (g \circ f)

If a monoidal context exists, include associator \alpha_{X,Y,Z} coherence.
Operational checks: normalize both bracketings and test equality up to declared isomorphisms. Record any need for explicit associators.

1.3 Naturality (when present)
Given a natural transformation \eta : F => G and any f : X -> Y,

- G(f) \circ \eta_X = \eta_Y \circ F(f)

Presence detection: only test when (F,G,\eta) are actually in scope.
Operational checks: construct commuting squares; compare normal forms of both paths.

1.4 Normalization hygiene
Typed terms/diagrams must satisfy:

- Beta: (\lambda x. b) a -> b[a/x]
- Eta (when declared): \lambda x. f x \simeq f (x not free in f)
- Alpha: renaming of bound variables is irrelevant
- Confluence: normalize(normalize(t)) = normalize(t) (idempotence)
- Compositionality: normalize respects composition (no cross-term artifacts)

Audit properties: strong normalization (no infinite rewrite), Church–Rosser (unique normal form up to alpha), and reproducible equality modulo declared isomorphisms only.

2) Witnesses and horns: observable participation in lawfulness

- Horn (obligation). A partially specified diagram (simplex/cube) that *must* admit a filler if the local laws hold.
- Witness (attestation). Concrete data (term/arrow/filler) that closes the horn and a certificate that the resulting diagram commutes (or the type checks).

Why it matters. Each filled horn + witness is *observed participation* in lawfulness. It is not a slogan; it is a checkable event.
Metrics.

- horns_total, horns_filled, fill_rate = horns_filled / horns_total
- witness_coverage = witnesses_attested / obligations_passed
- commute_failures (should be 0 in admissible histories)

Policy. No horn, no claim. No witness, no elevation. Proof rules remain the sole arbiters of correctness; invariants only *score* what proofs already certify.

3) Conservatism and determinism

3.1 Conservatism (no spurious laws)

- Never promote a pattern to a “law” unless it is forced by admissible proofs (types, horns, witnesses).
- Treat any cross-run regularity as *hypothesis* until backed by witnesses in the *current* context.
- Penalize overfitting to artifacts (encoding quirks, accidental equalities without witnesses).

3.2 Determinism for auditability

- Same inputs, same outputs, same normal forms, same filled horns, same logs.
- Record seeds, versions, and normalization traces.
- Re-runability is mandatory for public claims; stochasticity is disallowed in proof production (it is admissible only in *search order*, not in *proof content*).

4) Turning invariants into a measurable signal

Define a symbolic score R_estimate in [0,1] that *only* reflects invariant alignment of an admissible history H (it cannot override typing):

- pass_id(H) in {0,1} for identity/unit checks
- pass_assoc(H) in {0,1} for associativity checks
- pass_nat(H) in {0,1,undef} (ignored if naturals not present)
- norm_score(H) in [0,1] for hygiene (idempotence, confluence, compositionality)
- horn_quality(H) = fill_rate * witness_coverage

Aggregate (example; weights chosen a priori, preregistered):

- R_estimate(H) = w_id*pass_id + w_assoc*pass_assoc
    - w_nat*max(pass_nat,0) + w_norm*norm_score + w_h*horn_quality
      Normalize by (w_id + w_assoc + w_nat_active + w_norm + w_h).

Use in evaluation only:

- rank_by = S(H) + lambda * C_final(H,Lambda)
- where C_final can include a term proportional to R_estimate(H) and a term for MDL/description-length lift under a structural_prior.

5) Battery of invariant tests (quick operational spec)

1. Identity/unit tests. Sample typed f; verify id laws post-normalization.
2. Associativity tests. Compare both bracketings; audit need for \alpha insertions.
3. Naturality tests (if present). Build squares for random f; check commutation.
4. Normalization hygiene.
    - Idempotence: normalize(normalize(t)) == normalize(t).
    - Confluence spot-checks via parallel rewrites; require unique NF up to alpha.
    - Compositionality: normalize(g \circ f) equals normalized composite of normalized parts.
5. Horn suite. Generate obligations from local patterns; attempt fillers; verify witnesses; log failures.

Negative controls. Inject “twisted identities,” illegal bracketings, and bogus naturality squares to ensure the suite *catches* violations (guarding against false positives).

6) Edge handling (“when present” principle)

- Do not demand naturality or monoidal coherence where the corresponding structures are not declared.
- The test harness must introspect the current signature/context to enable only applicable invariants.

- Any “pass by absence” is recorded as “not applicable,” not as “true.”

Bottom line. Logos invariants are the fixed points: identity/unit, associativity, naturality (when present), and normalization hygiene. Witnessed horn fillings make conformity observable. Conservatism forbids law inflation; determinism guarantees auditability. Together, they provide a crisp, falsifiable signal of alignment with lawful structure—usable for ranking and analysis while leaving all proofs and rules untouched.

Mapping to ai1 (Equational Core) Without Changing It

Architectural separation. ai1 remains a closed equational kernel (types, normalization, horn obligations, witnesses, logs). The Logos layer is a read-only *evaluator*: it can rank and summarize ai1’s already-proved outputs, but it does not alter rules, proofs, or normal forms.

---

1) Typed normal forms → local “truth carriers”

Let NF(t) be the typed normal form of a term/diagram t : A -> B.

- Carrier properties.
  Idempotence: NF(NF(t)) = NF(t)
  Type preservation: if t : A -> B then NF(t) : A -> B
  Uniqueness (mod alpha/declared iso): two encodings reduce to the same NF iff they denote the same structure.
- What we store per carrier.
  (type_sig, nf_hash, invariants_fingerprint, provenance)
  where invariants_fingerprint records identity/unit, associativity, and hygiene checks that held *for this NF in context*.

- Why “truth carrier.”
  The NF is the smallest, reproducible representative of a local truth ai1 has *demonstrated*. It is the atom we rank later; we never mutate it.

---

2) Horns/witnesses → local resonance attestations

A horn obligation encodes a partial diagram that *must* admit a filler if local laws hold. A witness is concrete data that fills the horn and certifies commutation or type correctness.

- Attestation record.
  Horn k: (boundary_spec, filler_eta, cert)
  Status: passed/failed with normalized equality check(s).
- Local resonance signal.
  For each horn k, define r_k = 1 if passed with witness, else 0.
  Aggregate per history H:
  horn_fill_rate = sum_k r_k / #horns,
  witness_coverage = #witnessed / #passed.
- Policy. No witness, no elevation. Unwitnessed patterns may inspire *search*, but they do not count toward ranking.

---

3) Deterministic miner → honesty guardrails; no “weights,” no hallucination

- Determinism contract. Same inputs, same normalization rules, same seeds ⇒ identical NFs, horn outcomes, and logs. Any permitted parallelism must be order-canonicalized before emit.
- No-weights rule. ai1 uses no learned parameters and performs no gradient updates. Heuristics may affect search order only and must be fixed in advance (seeded, logged).

- No hallucination. Every emitted claim is backed by a type check, a normalization equality, or a witnessed horn. If a step is unverifiable, it is not emitted.

Audit fields (per run). ruleset_id, seed, env_fingerprint, time_window, input_set_hash, nf_hashes[], horn_outcomes[], log_checksum

---

4) Discovery-only stance: mining > training

- Mining (ai1). Explore the discrete space of well-typed expressions and diagrams; reduce to NF; pose horn obligations; attempt fillers; emit only *proved* artifacts. No parameter fitting.
- Training (not ai1). Fit parameters to data by loss minimization. Out of scope by design.

Minimal mining cycle (notation; not code).

inputs -> enumerate_well_typed()

-> normalize() -> NF_stream

-> pose_horns(NF_stream) -> obligations

-> attempt_fillers(obligations) -> witnesses

-> emit {NFs, horns, witnesses, logs}

Stopping conditions are structural (depth/size limits, rule budgets), not validation-loss plateaus.

---

5) How the Logos layer maps onto ai1 outputs (evaluation only)

Let Lambda be the fixed structural_prior (interpretive). For each admissible history H produced by ai1:

- Compute symbolic R_estimate(H) in [0,1].
  Example (weights preregistered):

- R_estimate = w_id*pass_id
- + w_assoc*pass_assoc
- + w_nat*pass_naturality_or_undef_as_0
- + w_norm*norm_hygiene_score
- + w_h*horn_fill_rate*witness_coverage
- normalize by sum of active weights

This consumes ai1's facts; it cannot override them.

- Compute C_final(H, Lambda) (end-coherence score).
  Components may include MDL lift under the structural prior and invariant-gap penalties. Again, read-only over ai1 outputs.
- Rank without feedback into proofs.
  S_total(H) = S(H) + \lambda * C_final(H, Lambda)
  Here S(H) is an effort/complexity proxy already logged by ai1. Ranking only affects which proved bundles we foreground to readers—never *what* counts as proved.

---

6) What gets published (the "admissible bundle")

For each selected H, publish:

- NFs: hashes + type signatures + normalization traces.
- Horns/Witnesses: obligations, fillers, certificates, equality checks.
- Invariant battery: pass/fail table (id/unit, assoc, naturality if present, hygiene).
- Scores: R_estimate, C_final, S(H), optional compression metrics K(*).
- Provenance: seeds, ruleset, environment, timestamps, log checksums. Everything necessary to fully replay the run and reproduce the same artifacts.

7) Failure handling and guardrails

- If a horn fails: record boundary, failing side's NF, and reason; no elevation.
- If hygiene fails (non-idempotent normalize, confluent breach): quarantine run; investigate before any ranking.
- If determinism drifts across replays: flag run as invalid; do not publish ranks.

---

8) Bottom line

- Typed normal forms are the portable atoms of truth ai1 already proves.
- Horns and witnesses are measurable participation in lawfulness.
- Deterministic mining is your honesty layer: no weights, no guessing, no hallucination.
- Discovery-only means we mine what is there; we do not *train* it into existence. The Logos evaluator merely orders the already-true—never invents it.

Operational Hook for ai2 (Optional, Non-intrusive)

Goal. Add a *read-only* evaluator to ai2 that re-ranks already-proved bundles without changing any proofs, rules, or normal forms.

1) Scoring (fixed, deterministic)

- Primary score:

score(H) := w1*comp2(H) + w2*r_estimate(H)

where:

  - comp2(H) = your existing compression/complexity utility from ai1/ai2 runs.

    - r_estimate(H) = symbolic overlap with Logos invariants (identity/unit, associativity, naturality_when_present, normalization_hygiene, horn_fill_rate, witness_coverage).

- r_estimate(H) construction (never overrides typing):

r_estimate := (w_id*pass_id*
*+ w_assoc*pass_assoc
+ w_nat*pass_nat_or_0*
*+ w_norm*norm_score
+ w_h*(horn_fill_rate*witness_coverage)) / Z

pass_* in {0,1}, norm_score in [0,1], Z = sum of active weights.

- Defaults (a priori, no training):
    - w1 = 1.0, w2 = 0.30
    - w_id = 1, w_assoc = 1, w_nat = 1, w_norm = 2, w_h = 2
    - If naturality is not present in context, set pass_nat_or_0 = 0 and subtract w_nat from Z.

Tie-breakers (deterministic): prefer lower S(H) (effort/action), then earlier timestamp, then lexicographic hash(nf_bundle).

2) Two-pass evaluation (separation of concerns)

Pass A: forward_search (ai1/ai2 kernel only)

- Enumerate well-typed candidates.
- Normalize to NF; pose horns; attempt fillers; emit witnesses.
- Produce admissible histories H with artifacts:
    - nf_hashes, type_sigs, normalization_traces
    - horn_outcomes (passed/failed), witnesses
    - comp2(H), S(H), logs, seeds, ruleset_id

Pass B: final_coherence_scoring (read-only)

- Compute invariant checks and r_estimate(H).
- Compute score(H) as above.
- Optionally compute c_final(H) for reports:

c_final(H) := a1*r_estimate(H) + a2*mdl_lift(H; structural_prior)

where mdl_lift = K(baseline) − K(with_structural_prior).
(Use a1=1.0, a2=0.0 by default to keep the minimal hook.)

- Rank admissible H by score(H).
- Emit only ranks and summaries; do not feed back into proofs.

3) Interfaces and fields (CSV/JSON)

Per history H, append:

- comp2, r_estimate, score
- pass_id, pass_assoc, pass_nat_or_0, norm_score
- horns_total, horns_filled, horn_fill_rate, witness_coverage
- s_action (S(H)), ruleset_id, seed, env_fingerprint, run_id

4) Reproducibility and guardrails

- Config hash: hash of {w1,w2,w_id,w_assoc,w_nat,w_norm,w_h}.
- Determinism: identical inputs + identical config_hash must yield identical ranks.
- Isolation: pass B reads artifacts only; it cannot invoke normalization or alter witnesses.
- “Never overrides typing”: any item failing typing/horn checks is excluded *before* scoring.

5) Controls and ablations (to prevent self-deception)

- Placebo prior: set r_estimate:=0 and confirm ranks reduce to comp2 only.
- Weight ablation: set w2:=0 and w1:=1 to verify pure comp2 ordering.
- Shuffle test: randomly permute invariant flags; rank quality should degrade.
- No effect on proofs: diff emitted proofs/logs with and without the hook must be empty.

6) Suggested directory and filenames (Windows)

- Input from ai1/ai2:
  C:\Users\doug\ai2\out\admissible\H_*.jsonl
- Scores out:
  C:\Users\doug\ai2\out\scores\ranks.csv
  C:\Users\doug\ai2\out\scores\by_history\H_*.json
- Config:
  C:\Users\doug\ai2\config\ai2_scores.config.json (weights, toggles)

7) Minimal run order (notation, not code)

init_config() -> load_weights()
H_stream := forward_search_emit_admissible() // ai1/ai2 kernel
for H in H_stream:
feats := compute_invariant_features(H)
r := r_estimate(feats)
s := w1*comp2(H) + w2*r
write_row(H, feats, r, s)
rank_all_by_score() -> publish_top_k()

8) What this buys you

- Fully deterministic, auditable prioritization of *proved* artifacts.
- A single scalar score that blends your comp2 with an invariant-based resonance proxy—without touching math, rules, or proofs.

- Clean on/off switch (w2=0) and falsifiable controls (placebo/shuffle/ablation).

Predictions and Tests (Falsifiable Directions)

Below are four concrete predictions with pre-registered measures, analysis plans, controls, and fail criteria. All math is ASCII-safe.

---

1) Compression advantage: Delta_K > 0 under Logos-prior encodings

Claim. Descriptions aligned with structural_prior (Lambda) compress better.
Metric. Delta_K := K(baseline_encoding) - K(prior_aligned_encoding) > 0.

Setup.

- Inputs: per history H, emit a canonical description file desc(H).
- Produce two byte strings:
  - desc_base := encode(desc(H)) using your standard schema.
  - desc_prior := encode_with_prior(desc(H), structural_prior=Lambda), e.g., factor out identity/unit, associativity, and hygiene as references, not repeated text.
- Approximate K(*) with a fixed compressor C (e.g., zstd level X), preregistered.

Test.

- For N independent histories, compute Delta_K_i.
- Primary test: one-sided Wilcoxon signed-rank or paired t-test on Delta_K_i, alpha=0.01.
- Effect size: mean(Delta_K_i)/sd(Delta_K_i). Report 95% CI.

Controls.

- Placebo prior: shuffle invariant tags; expect Delta_K ≈ 0.
- Compressor ablation: gzip vs zstd vs lzma; effect should persist.
- Adversarial check: remove structural_prior entirely; Delta_K must vanish.

Fail criterion. If median(Delta_K) ≤ 0 with power ≥ 0.9 (pre-computed), the compression prediction is not supported.

---

2) Negative-lag signal: transfer entropy future->present spikes at insight

Claim. Around discovery events, information about near-future traces helps explain present updates (boundary selection signature), without leakage.

Signals to log per step t.

- R_t (resonance), perf_t (performance), inv_flags_t (invariant vector), hash of NF bundle.
- discovery_event flags defined earlier (dR/dt >= tau_R AND dPerf/dt >= tau_P).

Measure.

- TE_future_to_present(tau) := transfer_entropy(X_t ; Y_{t+tau} | past), with tau < 0.
    - X_t := present variables (R_t, perf_t, inv_flags_t).
    - Y_{t+tau} := future summary (e.g., C_final at t+|tau|).
    - Use a fixed estimator (binning or kNN); preregister bins/k.

Test.

- Predefine a small set of negative lags, e.g., tau in {-3,-2,-1}.
- For windows [t* - W, t* + W] around each discovery_event t*, compute TE at taus and a baseline TE at tau=+1 (sanity).

- Primary: TE(tau<0) > TE_baseline using paired tests across events; alpha=0.01 after Bonferroni for |taus|.

Controls.

- Phase-randomize the future series (permute within window); TE should drop to baseline.
- “Leakage guard”: forbid any code path that reads future fields during Pass A; verify by code diff and hash.

Fail criterion. No significant TE increase at negative lags with preregistered estimator and controls ⇒ no support for boundary-selection signature.

---

3) Discovery clusters: epochal simultaneity vs. null Poisson baselines

Claim. Independent discoveries cluster in time beyond chance, consistent with global end-constraints.

Data.

- Define a discovery as any emitted bundle flagged discovery_event with witness_coverage >= theta and norm_score >= theta_norm.
- Aggregate across runs/days into timestamps {t_i}.

Models.

- Null: inhomogeneous Poisson with rate lambda(t) fitted to operational load (e.g., #candidates evaluated per hour).
- Alt: self-exciting or bursty process (e.g., Hawkes) OR piecewise-constant rate with change-points.

Test.

- Use a scan statistic over windows of size w in a preregistered grid (e.g., w in {30min, 1h, 2h}).

- Compute max excess E_w := max_over_windows [count_window - E_null_window].
- p-value via parametric bootstrap from the fitted inhomogeneous Poisson (10k draws).
- Report FWER-adjusted p across w.

Controls.

- Replace discovery_event with matched “pseudo-events” drawn from non-insight steps; clustering should disappear.
- Downsample to constant throughput to ensure load is not the cause.

Fail criterion. If observed clustering is consistent with null across all w after correction, claim not supported.

---

4) Ethics -> insight coupling: within-subject A/B on virtue/prayer protocols

Claim. Practices aimed at aligning with Logos (virtue/prayer) increase resonance and discovery likelihood.

Design.

- Within-subject randomized ABAB or BAAB blocks over D days.
- A-block: neutral control (quiet rest or secular reading).
- B-block: predefined virtue/prayer protocol (duration T, content fixed).
- Same work tasks, same code, no internet/news changes; seeds rotated deterministically per block.

Outcomes (pre-registered).

- Primary: discovery_rate := (#discovery_events per unit time).
- Secondary: mean dR/dt around attempts; mean comp2; witness_coverage.

Analysis.

- Mixed-effects: outcome ~ condition + (1|day).
- Report Cohen's d and 95% CI. Alpha=0.01.
- Sequential analysis allowed only if pre-declared.

Controls.

- Expectation control: include a neutral but time-matched mindfulness block in a three-arm sub-study (A vs B vs M) to separate generic focus from specifically virtue/prayer.
- Blinding: coder analyzing logs sees anonymized condition labels.
- Manipulation check: brief self-report scales (fixed, minimal) to confirm compliance.

Fail criterion. No significant lift in discovery_rate (and aligned secondary signals) with adequate power ⇒ coupling not supported.

Shared reproducibility rules

- All thresholds (tau_R, tau_P, theta, theta_norm), window sizes, compressors, TE estimators, and statistical tests are fixed in a pre-registration file and hashed (config_hash).
- No post-hoc metric swapping; exploratory results must be labeled as such.
- Every figure/table is re-generable from a single script that reads immutable logs.

Bottom line. Each prediction can be cleanly supported or refuted. If compression gains, negative-lag TE, clustering, or ethics-linked lifts fail robustly under controls, the Logos interpretation should be revised or rejected—while the equational core (ai1) stands on its own.

Case Notes and Measurement Plan

Purpose. Make every claim replayable end-to-end with fixed configs, immutable logs, and preregistered analyses.

---

1) What to log (granular, immutable)

Per time step t (or per candidate if stepwise isn't natural):

- run_id, history_id (H), step_index=t, utc_timestamp (ISO-8601, Z)
- ruleset_id, seed, config_hash, env_fingerprint (OS, Python, CPU model)
- psi_hash_t (hash of state rep), nf_count_t, horns_posed_t, horns_filled_t
- pass_id_t, pass_assoc_t, pass_nat_t (0/1/undef), norm_score_t in [0,1]
- comp2_t, R_t (if computed online), performance_t
- discovery_flag_t (per spec: dR/dt>=tau_R & dPerf/dt>=tau_P)

Per admissible history H (bundle level):

- nf_bundle: [(type_sig, nf_hash, norm_trace_hash) …]
- horn_bundle: [(horn_id, boundary_hash, filler_hash, cert_hash, pass) …]
- invariants_fingerprint: {id, assoc, nat_or_undef, hygiene_score}
- aggregates: horns_total, horns_filled, horn_fill_rate, witness_coverage
- comp2(H), S(H), R_estimate(H), C_final(H,Lambda), score(H)
- desc_base_len, desc_prior_len (for MDL tests), Delta_K
- provenance: input_set_hash, start_utc, end_utc, log_checksum

File hygiene. All logs are append-only JSONL/CSV with SHA256 trailer (log_checksum). Timestamps are UTC; local time never used in analysis.

---

2) How to compute R_estimate reproducibly

Definition (fixed before runs):

- R_estimate(H) = (w_id*pass_id + w_assoc*pass_assoc + w_nat*pass_nat_or_0
    - w_norm*norm_score + w_h*horn_fill_rate*witness_coverage) / Z, where Z = sum of active weights.

Reproducibility rules:

- Weights {w_id,w_assoc,w_nat,w_norm,w_h} and "active" set are stored in ai2_scores.config.json; config_hash := sha256(canonical_json(config)).
- pass_nat_or_0 = 0 when naturality not declared "present"; do not silently default to 1.
- norm_score is a deterministic function of normalization tests (idempotence, confluence spot-checks, compositionality), each with fixed seeds and case lists hashed as norm_suite_hash.
- No stochastic elements; if any sampling is required (e.g., selecting test morphisms), PRNG is seeded by (run_id, history_id, ruleset_id).

---

3) How to compute C_final reproducibly

Minimal form:

- C_final(H,Lambda) = a1*R_estimate(H) + a2*MDL_lift(H;Lambda).

MDL_lift protocol (fixed):

- desc_base := canonical_encode(desc(H)) with schema v1.
- desc_prior := same schema but factoring declared invariants via references.
- Compression C is fixed (e.g., zstd vX, level L).
- K(x) := |C(x)| in bytes. MDL_lift := K(desc_base) − K(desc_prior).

- Exclude volatile fields (timestamps, hashes) from desc(*); schema lists fields explicitly. Schema and compressor version are recorded in config_hash.

If a2=0 (default minimal hook), MDL path is dormant but remains definable.

---

4) Pre-registration blueprint (single immutable file)

Create prereg_YYYYMMDD.json with:

- hypotheses: H1..Hk mapped to sections (Delta_K>0, TE neg-lag, clustering, ethics A/B)
- metrics & thresholds: tau_R, tau_P, theta, theta_norm, window sizes, TE estimator (bins or kNN with k), compressors and levels, significance alpha, correction plan
- analysis plan: exact tests (paired t or Wilcoxon; bootstrap count; Hawkes vs Poisson scan)
- inclusion/exclusion: what counts as admissible H; how to handle undef naturality
- stopping rules & multiple-testing correction
- power targets and planned N (see below)
- freeze: prereg_hash := sha256(canonical_json(prereg))

The prereg_hash is embedded in every run header and in all figures/tables.

---

5) Power analyses (planning, not post-hoc)

Rules of thumb (paired designs; two-sided unless noted):

- Compression (Delta_K): expecting small-to-moderate d=0.35. For power=0.9, alpha=0.01 → N≈240 histories. If d=0.5, N≈120. Use the larger N if unsure.

- TE negative-lag: effect is typically small. With windowed TE and Bonferroni over 3 lags, plan ≥150 discovery windows. Pilot to estimate variance; if sd is high, double windows.
- Clustering: choose rate-matched inhomogeneous Poisson. With 10k bootstraps and 3 window sizes, plan ≥300 discovery events to detect moderate excess with FWER 0.01.
- Ethics A/B (within-subject): target d=0.4. ABAB or BAAB with 16 blocks total (8 per condition) often suffices; if blocks are short, increase to 24 blocks.

Document assumptions and recalc N after a small pilot; update only via a new prereg with a new prereg_hash.

---

6) Failure modes and mitigations

- Leakage across passes. Any read of future fields during Pass A invalidates TE tests. Mitigation: static code scan + runtime guard that hashes Pass A outputs before Pass B runs.
- Determinism drift. Non-canonical parallelism or time-dependent seeds change outputs. Mitigation: canonicalize ordering; seeds = sha256(run_id, ruleset_id, history_index); replay checker.
- “Pass-by-absence” inflation. Counting absent naturality as success. Mitigation: nat_or_undef must be logged explicitly; only “present & passed” counts.
- MDL double-counting. Prior encoding smuggles results (e.g., repeats the answer). Mitigation: prereg schema forbids including computed summaries; audits compare field lists.
- Poisson mis-specification. Throughput variations fake clustering. Mitigation: fit inhomogeneous rate to observed throughput; downsample to constant-rate control.

- TE estimator bias. Choice of bins/k inflates TE. Mitigation: prereg single estimator; show placebo (phase-randomized) TE near zero.
- Normalization hygiene gaps. Non-idempotent normalize or confluent breaches corrupt nf hashes. Mitigation: quarantine run; hygiene suite must pass before any scoring.
- P-hacking drift. Mid-run threshold changes. Mitigation: thresholds live only in prereg; config_hash mismatch aborts analysis.

---

7) Case-note template (per notable run)

CaseID, hypothesis_ids, brief narrative (what changed), links to:

- top_k histories (by score), their artifacts, and replication script id
- anomalies (failed horns, hygiene flags), and adjudication notes
- decision: support/neutral/refute for each hypothesis (with p-values, CIs)

---

8) Directory & filenames (Windows)

- Inputs from kernel: C:\Users\doug\ai2\out\admissible\H_*.jsonl
- Scores: C:\Users\doug\ai2\out\scores\ranks.csv and by_history\H_*.json
- TE/MDL analyses: C:\Users\doug\ai2\analysis\te\..., \mdl\...
- Prereg & configs: C:\Users\doug\ai2\config\prereg_YYYYMMDD.json, ai2_scores.config.json

Bottom line. Fixed schemas, hashed configs, and preregistered analyses turn every "insight" into a replayable, falsifiable claim—while leaving the equational core untouched.

You said:

Expand: Objections & Replies "Theology in science?" → interpretive layer ≙ structural prior; testable edges. "Teleology smuggling?" → constrained to

evaluation order, not proof rules. “Retrocausality paradoxes?” → boundary optimization, not message sending. “Just-so prior?” → predictions in §8 distinguish it from baselines.

Objections & Replies

1) “Theology in science?”

Objection. Invoking “Logos” imports theology into a technical workflow.
Reply. “Logos” is used as an interpretive label for a structural prior. Every technical claim is expressed without theological vocabulary by substituting structural_prior = Lambda.
Operational guardrails. The equational core (types, NFs, horns, witnesses) is unchanged. The Logos layer is read-only ranking: toggle w2=0 to remove it entirely.
Testable edges. Predictions that *must* hold if the layer is meaningful: Delta_K > 0 under prior-aligned encodings; negative-lag transfer entropy near discovery events; clustering beyond rate-matched Poisson; ethics→insight lift (see “Predictions & Tests”).
Fail case. If ablations (placebo prior, shuffled flags, w2=0) erase effects, or effects never appear with adequate power, the interpretation is rejected while ai1 remains intact.

---

2) “Teleology smuggling?”

Objection. Final causes are being sneaked into the proof system.
Reply. Teleology is confined to evaluation order, not proof rules. Proofs are produced exactly as before; the two-pass design only re-ranks already-proved histories by score = w1*comp2 + w2*R_estimate.
Operational guardrails. No feedback from ranks into derivations; Pass A (proof) is completed and hashed before Pass B (ranking) runs. Deterministic replays must produce identical proofs with or without the ranking pass.

Testable edges. Show a diff: proofs/logs are byte-identical across runs with w2=0 vs w2>0. Any difference = design violation.

---

3) “Retrocausality paradoxes?”

Objection. If ends influence selections, do we get messaging from the future or paradoxes?
Reply. We use boundary optimization, not signaling. Final constraints help select globally consistent histories; they do not transmit information backward in time.
Operational guardrails. Pass A cannot read any future fields; its outputs are hashed before Pass B. TE analysis includes phase-randomization controls to ensure any negative-lag signal isn’t leakage.
Testable edges. If code audit or hash checks reveal future reads in Pass A, results are void. If negative-lag TE persists only without controls, the claim fails.

---

4) “Just-so prior?”

Objection. A hand-picked prior can explain anything post hoc.
Reply. The prior is minimal and preregistered (identity/unit, associativity, naturality-when-present, normalization hygiene, horn quality). No tuning to outcomes.
Operational guardrails. Pre-registration hash (prereg_hash) freezes metrics/thresholds. Controls: placebo prior (all zeros), weight ablation (w2=0), shuffle tests on invariant flags, compressor ablations for MDL.
Testable edges. Distinctive predictions (compression lift, TE negative-lag, clustering, ethics→insight) must beat baselines with corrected α. If not, abandon the prior—or keep it only as metaphor.

Additional common objections (and compact replies)

“Isn’t this just Bayesianism with a fancy name?”
Reply: Close in spirit to a *fixed* structural prior, but no parameter learning and no

posterior over weights; ai1 mines proofs, not models. The prior affects only ranking and MDL accounting, not derivations.

“Underdetermination: many priors fit.”
Reply: Good—compare them. Pre-register competing minimal priors and evaluate out-of-sample. The winning prior must outperform on *all* preregistered tests, not just one.

“Ethics→insight is confounded by expectation/placebo.”
Reply: Within-subject ABAB, blinding of analysts, and a third mindfulness arm control for expectancy. Failure under these controls falsifies the coupling.

“Compression ≠ truth.”
Reply: Agreed; that’s why compression is one metric, never a proof. Only witnessed, well-typed results are admissible; compression helps rank admissible bundles, not certify them.

“You’ll over-reward simplicity.”
Reply: Witnesses and horn obligations prevent collapsing non-isomorphic structures. MDL lift is reported alongside invariants; contradictions or type violations are excluded before scoring.

“MDL is circular if the schema bakes in the answer.”
Reply: The schema is fixed and hashed pre-run; it forbids including computed summaries as fields. Placebo schemas must erase the Delta_K effect.

Bottom line. The layer is interpretive, optional, and falsifiable. It never changes proofs; it just proposes measurable regularities. If those regularities don’t survive ablations and controls, we drop the interpretation and keep the math.

Ethical Posture and Epistemic Humility

1) Participation, not possession

- Thesis. Knowledge is participation in pre-existing structure, not ownership of it.
- Resonance ≠ deification. Our "resonance" quantities (R, R_estimate) are alignment metrics, not moral worth, sanctity, or authority. High R says "this artifact coheres with declared invariants," not "this artifact (or its author) is divine."
- Two dialects, one core. Readers may substitute "structural_prior" for "Logos" with no loss to any technical claim. The interpretive layer never confers spiritual status on proofs or people.

2) Practical commitments (how humility is enforced)

- Determinism by design. Same inputs ⇒ same NFs, same horn outcomes, same logs. No hidden randomness in proof production.
- Witness or withhold. Every promoted claim is backed by a witnessed horn or typed normalization equality; otherwise it is logged as exploratory only.
- No hallucination pledge. If an artifact cannot be verified by the kernel, it is not emitted as truth.
- Pre-registration & all-results reporting. Hypotheses, thresholds, estimators, and analysis scripts are frozen (config_hash, prereg_hash). We publish wins, nulls, and failures.
- Ablations on by default. Placebo priors, weight zeroing (w2=0), and shuffle tests are run and reported with the same prominence as headline results.

3) Why transparency matters for public trust

- Determinism ⇒ reproducibility. Anyone can replay a run and obtain byte-identical proofs/logs. This resists authority-by-rhetoric.

- Witnesses ⇒ accountability. Horn fillings and certificates make “lawfulness” observable rather than asserted.
- Complete provenance. Seeds, ruleset_id, environment fingerprints, and checksums are exported with artifacts so third parties can audit pipeline integrity.
- Clear separations. Pass A (proof) is sealed before Pass B (ranking). Any deviation invalidates claims tied to the interpretive layer.

4) Misuse guardrails

- No ranking of persons. Resonance metrics apply to artifacts (histories, NFs), never to humans, groups, or beliefs.
- No policy laundering. Logos language cannot be cited as a warrant for social or political authority. Technical results stand on proofs, not metaphors.
- No surveillance/credit scoring. We prohibit applying R or C_final to individual behavior datasets. The system is for equational artifacts only.
- Schema discipline. MDL encodings may not smuggle conclusions; schemas are preregistered and audited.

5) Handling failure (and being glad for it)

- If preregistered predictions (ΔK>0, negative-lag TE, clustering, ethics→insight) do not replicate under controls, we withdraw or revise the interpretive claims. The equational core remains valid and useful regardless.
- Negative results are archived with DOIs/hashes so others can learn from our misses.

6) Communication norms

- Avoid “proof of God” rhetoric. We present a testable interpretation, not a theological verdict.
- Explicitly label the Logos layer as optional and non-binding for adoption of the core methods.

- When presenting to broad audiences: lead with proofs/witnesses; place interpretation in a clearly marked sidebar.

7) Ethics checklist (to include with each release)

- Deterministic kernel; replay script produces identical outputs
- Horn obligations + witnesses published; failures documented
- Prereg hash + config hash included; thresholds and tests frozen
- Ablations (placebo prior, w2=0, shuffle) reported alongside results
- No human-subject data used (or IRB/consent attached if applicable)
- No claims about persons or groups; artifacts only
- Clear disclaimer: “Interpretive layer optional; does not affect proofs”

Bottom line. Our ethos is truth without triumphalism: proofs where possible, witnesses where needed, data when disputed, and quiet retraction when wrong. Participation in intelligibility is a privilege; claiming to possess it is a category error.

Related Work (Pointers)

Below are compact entry points—grouped by theme—with representative authors/works you can cite. They’re pointers, not an exhaustive bibliography.

1) Discovery vs. invention (mathematics & science)

- Mathematical realism / Platonism. Plato; Gödel (“What is Cantor’s Continuum Problem?”), Penrose (*The Road to Reality*; *Shadows of the Mind*), Balaguer (*Platonism and Anti-Platonism in Mathematics*), Shapiro (*Philosophy of Mathematics: Structure and Ontology*), Maddy, Colyvan.
- Nominalism / anti-realism. Field (*Science Without Numbers*), Putnam (various essays), Benacerraf (“Mathematical Truth”), Hellman (modal structuralism).

- Structural realism (philosophy of science). Worrall ("Structural realism: the best of both worlds?"), Ladyman & French (*Every Thing Must Go*).
- Convergence & "unreasonable effectiveness." Wigner (essay), Tegmark (Mathematical Universe Hypothesis), Lakatos (*Proofs and Refutations*), Merton (multiples in discovery).

2) Boundary-condition / time-symmetric physics

- Variational principles & paths. Hamilton's principle; Feynman (path integrals).
- Absorber / transactional ideas. Wheeler & Feynman (absorber theory); Cramer (Transactional Interpretation).
- Two-state / time-symmetric QM. Aharonov, Bergmann & Lebowitz (ABL rule); Aharonov & Vaidman (two-state vector formalism).
- Retrocausality without signaling. Huw Price (*Time's Arrow & Archimedes' Point*), Leifer & Pusey (time-symmetric quantum theory), Wharton (all-at-once/boundary constraints), Sutherland (causally symmetric models).

3) Guiding-wave (pilot-wave) approaches

- Foundations. de Broglie (pilot idea), Bohm (1952 hidden-variables), Bohm & Hiley (*The Undivided Universe*).
- Mathematical development. Dürr, Goldstein & Zanghì (Bohmian mechanics), Holland (*The Quantum Theory of Motion*).
- Extensions / nonequilibrium. Valentini (subquantum H-theorem, signal locality restoration), Sutherland (time-symmetric Bohm variants).
- Comparative reviews. Goldstein (survey articles on Bohmian mechanics).

4) Interpretability, compression, and MDL/K-complexity

- Algorithmic foundations. Solomonoff (universal induction), Kolmogorov (complexity), Chaitin; Li & Vitányi (*An Introduction to Kolmogorov Complexity and Its Applications*).

- Minimum Description Length (MDL). Rissanen (original MDL papers), Grünwald (*The Minimum Description Length Principle*).
- Universal AI / compression-as-understanding. Hutter (*Universal AI*), Schmidhuber (compression progress, discovery as compression).
- Information bottleneck & sparsity. Tishby, Pereira & Bialek (IB); Shwartz-Ziv & Tishby (deep nets as information bottlenecks).
- Mechanistic interpretability (circuits). Olah et al. ("Circuits," "Zoom In" series); follow-ups on feature superposition and toy models.

5) Category theory / HoTT context (for horns, witnesses, normalization)

- CT foundations. Mac Lane (*Categories for the Working Mathematician*), Awodey (*Category Theory*).
- HoTT & univalence. *Homotopy Type Theory: Univalent Foundations of Mathematics* (The HoTT Book); Voevodsky (univalence papers).
- Horns & fillers (simplicial sets). Quillen (model categories), Goerss & Jardine (*Simplicial Homotopy Theory*), Hatcher (intro homotopy).
- Normalization & confluence. Church, Rosser (confluence), Tait (strong normalization), Barendregt (*Lambda Calculus*), Harper (type theory texts).

How to use these pointers.

- For "discovery vs invention," pair a realist (Gödel/Penrose) and an anti-realist (Field/Benacerraf) to frame both sides.
- For time symmetry, cite one principle path source (Feynman) plus one boundary-constraint advocate (Price/Wharton) and one no-signaling retro model (Aharonov or Leifer & Pusey).
- For guiding-wave, contrast Bohm/Hiley with a concise Goldstein survey.
- For compression/interpretability, anchor with Solomonoff–Kolmogorov–Rissanen, then add one modern IB or circuits reference.

- For CT/HoTT, use Mac Lane + HoTT Book as canonical anchors; add a horns/fillers reference to justify your “witnessed horn” vocabulary.

Limitations and Future Work

1) Formalize inner products over symbolic states; refine C_final

Limitation. Our resonance term R currently assumes an abstract inner product on representations of states and the prior; without a fixed construction, R risks encoding dependence.

Plan. Make the representation and kernel explicit and positive-definite.

- Representation (Phi_map). Map a typed artifact or history H to a feature bundle:
    - invariant bag: counts/booleans for id/unit, assoc, (when present) naturality, hygiene tests
    - horn–witness graph: nodes = obligations; edges = shared boundaries; labels = pass/fail
    - normalization trace sketch: multiset of rewrite steps
    - type signature sketch: normalized arity/shape descriptors
      All features are deterministic, schema-versioned, and hashed.
- Kernels (choose 1–2, preregister):
    - K_inv((psi,Lambda)) = cosine over invariant vectors
    - K_hw((psi,Lambda)) = normalized kernel over horn–witness graphs (e.g., Weisfeiler–Lehman subtree counts)
    - K_mix = alpha*K_inv + (1−alpha)*K_hw with fixed alpha in prereg (no fitting)

- Inner product and resonance.
    - Let K be the chosen PD kernel. Define
      R := |K(psi, Lambda)|^2 / (K(psi,psi) * K(Lambda,Lambda)) in [0,1].
    - Prove positivity of K (or cite standard results) and log kernel_id + alpha.
- Refine the end-coherence functional.
    - C_final(H,Lambda) = a1*R_estimate(H) + a2*MDL_lift(H;Lambda)
        - a3*(−InvGap(H;Lambda)) + a4*Stability(H) + a5*Universality(H).
    - InvGap: sum_j max(0, violation_j − tol_j) over applicable invariants.
    - Stability: agreement of invariant outcomes across alternative encodings (e.g., 1 − Hamming_distance / length).
    - Universality: re-run the same pattern in minimally different contexts (renamings, iso-equivalences) and score reproducible success.
      All weights a* are fixed in prereg (no tuning to outcomes).

Risks. Goodharting on the kernel; codec-dependence in MDL. Mitigations. Placebo kernels/codecs in ablations; require effects to persist across 2+ kernels and 2+ compressors.

---

2) Broaden invariant sets; stress-test against adversarial seeds

Limitation. Current “Logos invariants” are minimal. This can under-penalize near-miss structures or over-credit simple cases.

Plan. Expand the “when present” test battery and build an adversarial fuzzer.

- Invariants to add (enabled only if declared in context):
    - functoriality: F(id)=id, F(g∘f)=F(g)∘F(f)
    - triangle/pentagon coherence where monoidal structure is present

    - adjunction sanity: unit/counit triangle laws when adjoints are declared
    - (co)limit spot checks: pullback/pushout commutation in small diagrams
    - naturality families: batch squares for randomly chosen f: X->Y
    - homotopy-level checks (if HoTT context declares them): basic horn-filler axioms in simplicial analogs
- Adversarial seed suite (type-preserving mutations):
    - twisted identities (id' that fails on one side), non-associative bracket traps
    - alpha-eq/name-capture booby traps; ghost witnesses (missing certificates)
    - iso smuggling (declaring isomorphisms not witnessed)
    - degenerate horns that "almost commute" within tolerance
      Mutators are deterministic (seeded by run_id, ruleset_id) and logged.
- Targets. Drive false-accept rate of invariants below 1e−6 on synthetic adversarials; demonstrate that R_estimate does not inflate under adversarial pressure.

Risks. Over-constraint leading to excessive false negatives. Mitigations. Strict "when present" gating; document tol_j per invariant; report both precision and recall on a labeled synthetic suite.

---

3) Bridge to empirical cognition studies on "insight" events

Limitation. Our negative-lag and resonance claims are system-internal. Without human data, they risk being dismissed as implementation artifacts.

Plan. Design preregistered, low-burden studies that align machine "discovery events" with human "aha" signatures—correlational, not medical.

- Tasks. Proof sketches, equivalence spotting, or diagram completion with CT/HoTT-literate participants. Log keystrokes, timestamps, and self-marked "aha" pings (single keypress).
- Signals (minimal instrumentation first).
  behavioral: latency distributions, revision bursts, compression proxies (bytes in participant's evolving sketch);
  optional lab add-ons in later phases: eye-tracking; EEG/fNIRS if IRB-approved.
- Analyses (preregistered).
  - Transfer entropy: TE(future_edits -> present_action | past) at negative lags around self-marked insight; placebo via phase randomization.
  - Compression proxy shift: Delta_K of the participant's sketch before/after "aha".
  - Convergence: inter-subject simultaneity of breakthroughs under fixed prompts.
- Ethics & privacy. Artifact-level evaluation only; no ranking of persons; anonymization; IRB where required; open materials and prereg on OSF-like platform.

Risks. Expectation/placebo effects, lab-specific artifacts. Mitigations. ABAB blocks (e.g., neutral vs contemplative preparation), blinded analysis labels, multi-site replication.

---

4) Additional threats to validity (and counters)

- Encoding dependence. Results hinge on Phi_map. → Use two dissimilar kernels; require replication under both.
- Hidden stochasticity. Non-canonical parallelism changes outputs. → Canonicalize order; replay checksums required.

- Schema leakage in MDL. Prior codec encodes the answer. → Fixed schema forbids computed summaries; placebo schema ablation.
- Throughput confounds in clustering. → Fit inhomogeneous Poisson to observed load; downsample control.
- Estimator sensitivity in TE. → Fix estimator/bins; report identical analysis under kNN and binned variants; placebos must go to ~0.
- Overreach of interpretation. → Logos remains optional; if predictions fail with power, retire or revise interpretation while keeping the equational core.

---

5) Roadmap (concrete deliverables)

- Phase I (kernel formalization, 4–6 weeks): spec Phi_map v1; prove PD for K_inv/K_hw; implement kernel_id switch; wire R and C_final with fixed weights; publish schema and replay script.
- Phase II (invariants + adversarials, 6–8 weeks): extend test battery; ship deterministic mutator/fuzzer; report precision/recall; set tol_j; update R_estimate accordingly.
- Phase III (cognition bridge, pilot 8–12 weeks): prereg tasks; collect N≈20 pilots; analyze TE and Delta_K proxies; decide on multi-site replication.

Bottom line. Formal kernels make R principled; richer invariants and adversarial fuzzing harden R_estimate; careful human studies test whether the system's "resonance" co-moves with genuine insight—moving the proposal from metaphor toward measurement.

References

Youvan, D. C. (2025). Achieving Artificial Intelligence (youvan.ai1) on a PC with Category Theory & HoTT. ResearchGate preprint, September 2025. DOI: 10.13140/RG.2.2.36085.03047

Aharonov, Y., Bergmann, P. G., & Lebowitz, J. L. (1964). Time symmetry in the quantum process of measurement. *Physical Review*, 134(6B), B1410–B1416.

Aharonov, Y., & Vaidman, L. (2008). The two-state vector formalism: An updated review. In *Time in Quantum Mechanics* (Vol. 1, pp. 399–447). Springer.

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

Balaguer, M. (1998). *Platonism and Anti-Platonism in Mathematics*. Oxford University Press.

Barendregt, H. (1984). *The Lambda Calculus: Its Syntax and Semantics* (rev. ed.). North-Holland.

Benacerraf, P. (1973). Mathematical truth. *The Journal of Philosophy*, 70(19), 661–679.

Bohm, D. (1952). A suggested interpretation of the quantum theory in terms of "hidden" variables. I & II. *Physical Review*, 85(2), 166–193; 180–193.

Bohm, D., & Hiley, B. J. (1993). *The Undivided Universe: An Ontological Interpretation of Quantum Theory*. Routledge.

Chaitin, G. J. (1975). A theory of program size formally identical to information theory. *Journal of the ACM*, 22(3), 329–340.

Church, A., & Rosser, J. B. (1936). Some properties of conversion. *Transactions of the American Mathematical Society*, 39(3), 472–482.

Cramer, J. G. (1986). The transactional interpretation of quantum mechanics. *Reviews of Modern Physics*, 58(3), 647–688.

Dürr, D., Goldstein, S., & Zanghì, N. (1992). Quantum equilibrium and the origin of absolute uncertainty. *Journal of Statistical Physics*, 67(5–6), 843–907.

Feynman, R. P., & Hibbs, A. R. (1965). *Quantum Mechanics and Path Integrals*. McGraw-Hill. (Emended ed. by F. Styer, 2010, Dover.)

Field, H. (1980). *Science Without Numbers: A Defence of Nominalism*. Princeton University Press.

Gödel, K. (1964). What is Cantor's continuum problem? In *Philosophy of Mathematics* (pp. 258–273). Prentice-Hall.

Goerss, P. G., & Jardine, J. F. (1999). *Simplicial Homotopy Theory*. Birkhäuser.

Grünwald, P. D. (2007). *The Minimum Description Length Principle*. MIT Press.

Harper, R. (2016). *Practical Foundations for Programming Languages* (2nd ed.). Cambridge University Press.

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

Holland, P. R. (1993). *The Quantum Theory of Motion: An Account of the de Broglie–Bohm Causal Interpretation of Quantum Mechanics*. Cambridge University Press.

Hutter, M. (2005). *Universal Artificial Intelligence: Sequential Decisions based on Algorithmic Probability*. Springer.

Kolmogorov, A. N. (1965). Three approaches to the quantitative definition of information. *Problems of Information Transmission*, 1(1), 1–7.

Ladyman, J., & Ross, D., with Spurrett, D., & Collier, J. (2007). *Every Thing Must Go: Metaphysics Naturalized*. Oxford University Press.

Lakatos, I. (1976). *Proofs and Refutations*. Cambridge University Press.

Leifer, M. S., & Pusey, M. F. (2017). Is a time symmetric interpretation of quantum theory possible without retrocausality? *Proceedings of the Royal Society A*, 473(2202), 20160607.

Li, M., & Vitányi, P. (2008). *An Introduction to Kolmogorov Complexity and Its Applications* (3rd ed.). Springer.

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

Maddy, P. (1997). *Naturalism in Mathematics*. Oxford University Press.

Merton, R. K. (1961). Singletons and multiples in scientific discovery. *Proceedings of the American Philosophical Society*, 105(5), 470–486.

Penrose, R. (2004). *The Road to Reality*. Jonathan Cape.

Price, H. (1996). *Time's Arrow and Archimedes' Point*. Oxford University Press.

Quillen, D. (1967). Homotopical algebra. *Lecture Notes in Mathematics* (Vol. 43). Springer.

Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, 14(5), 465–471.

Schmidhuber, J. (2009). Simple algorithmic theory of discovery, novelty, curiosity, and creativity. *IEEE Transactions on Autonomous Mental Development*, 2(3), 230–247. (Early tech reports 1991–2002.)

Schreiber, T. (2000). Measuring information transfer. *Physical Review Letters*, 85(2), 461–464.

Shapiro, S. (1997). *Philosophy of Mathematics: Structure and Ontology*. Oxford University Press.

Shwartz-Ziv, R., & Tishby, N. (2017). Opening the black box of deep neural networks via information. arXiv:1703.00810.

Solomonoff, R. J. (1964). A formal theory of inductive inference, parts I and II. *Information and Control*, 7(1), 1–22; 7(2), 224–254.

Sutherland, R. I. (2006). Causally symmetric Bohm model. *Studies in History and Philosophy of Modern Physics*, 37(3), 378–397.

Tegmark, M. (2008). The mathematical universe. *Foundations of Physics*, 38(2), 101–150.

The Univalent Foundations Program. (2013). *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study.

Tishby, N., Pereira, F. C., & Bialek, W. (1999). The information bottleneck method. *Allerton Conference on Communication, Control, and Computing*.

Valentini, A. (1991). Signal-locality, uncertainty, and the subquantum H-theorem. I & II. *Physics Letters A*, 156(1–2), 5–11; 158(1–2), 1–8.

Wheeler, J. A., & Feynman, R. P. (1945). Interaction with the absorber as the mechanism of radiation. *Reviews of Modern Physics*, 17(2–3), 157–181. (See also 1949 follow-up.)

Wigner, E. P. (1960). The unreasonable effectiveness of mathematics in the natural sciences. *Communications on Pure and Applied Mathematics*, 13(1), 1–14.

Worrall, J. (1989). Structural realism: The best of both worlds? *Dialectica*, 43(1–2), 99–124.

Olah, C., et al. (2020). Zoom In: An introduction to circuits. *Distill*. (Circuits thread: 2020–2021.)

Appendix A: ASCII-Safe Formal Schema

This appendix fixes names, objects, and computations in plain ASCII so every reader can reproduce numbers from logs without changing ai1's proofs.

A1. Variable glossary (symbols are ASCII identifiers)

- H := a candidate history (proof trace / output bundle)
- t := discrete step index (t = 0,1,2,...)
- psi := agent/system state at step t (abstract)
- Lambda := structural_prior (optional "Logos prior")
- Phi_map(.) := deterministic encoder from objects to a feature space V
- phi_psi := Phi_map(psi)
- phi_L := Phi_map(Lambda)
- K(x,y) := positive-definite kernel on encoded objects
- ip(u,v) := inner product in V (when using explicit vectors)
- norm(u) := sqrt(ip(u,u))
- R := resonance scalar in [0,1]
- Psi_L(.) := nonnegative "intelligibility density" induced by Lambda in V
- V_Lambda(.) := guidance potential (heuristic; not used in proofs)
- S(H) := baseline cost/action for H (effort, size, etc.)
- C_final(H,L) := end-coherence score for H w.r.t. Lambda
- S_total(H) := S(H) + lambda * C_final(H,Lambda), lambda >= 0
- comp2(H) := existing compression/complexity utility from ai1/ai2
- inv flags := pass_id, pass_assoc, pass_nat_or_0 in {0,1}
- norm_score := normalization hygiene score in [0,1]

- horn stats := horns_total, horns_filled, horn_fill_rate, witness_coverage
- R_estimate := symbolic overlap with invariant battery in [0,1]
- score(H) := w1*comp2(H) + w2*R_estimate(H), read-only ranking
- MDL_lift := K_len(desc_base) - K_len(desc_prior) using fixed compressor
- config_hash := sha256(canonical_json(config))
- prereg_hash := sha256(canonical_json(prereg))

A2. Norms and inner-products (two interchangeable realizations)

A2.1 Vector inner product (explicit feature vectors)

- Build phi_psi, phi_L as fixed-length real vectors via Phi_map.
- Define:
    - ip(u,v) := sum_i u[i]*v[i]
    - norm(u) := sqrt(ip(u,u))
    - cos(u,v) := ip(u,v) / (norm(u)*norm(v) + eps)
    - R := (cos(phi_psi, phi_L))^2 in [0,1]
- eps is a small positive constant to avoid division by zero; log eps in config.

A2.2 Kernel inner product (implicit feature space)

- Choose a PD kernel K; examples (fixed before runs):
    - K_inv := cosine over invariant vectors only
    - K_hw := WL-style subtree count kernel over horn–witness graphs
    - K_mix := alpha*K_inv + (1-alpha)*K_hw with fixed alpha in config
- Define normalized similarity:
    - cos_K(psi,L) := K(psi,L) / sqrt(K(psi,psi)*K(L,L) + eps)
    - R := (cos_K(psi,L))^2

Both realizations must produce the *same* R within tolerance if K is the dot-product on the same vectors.

A3. Definitions (R, V_Lambda, S_total)

A3.1 Resonance

- R := |<phi_psi, phi_L>|^2 when ip is defined directly, or
- R := (K(psi,Lambda))^2 / (K(psi,psi)*K(Lambda,Lambda) + eps) using kernel K.
- Range: R in [0,1].

A3.2 Guidance potential (heuristic; optional)

- V_Lambda(phi) := - grad_phi(log(Psi_L(phi) + eps)).
- Discrete step (if enabled; eta >= 0 fixed in config):
    - phi_psi_next := Normalize(phi_psi - eta * V_Lambda(phi_psi)).
- If guidance is disabled, set eta := 0. Never used to alter proofs.

A3.3 Boundary objective

- S_total(H) := S(H) + lambda * C_final(H,Lambda), lambda >= 0.
- Minimal C_final:
    - C_final(H,Lambda) := a1*R_estimate(H) + a2*MDL_lift(H;Lambda).
- All weights (lambda, a1, a2) are fixed in config; no fitting to outcomes.

A4. R_estimate construction (symbolic, proof-respecting)

Given an admissible H (typed, witnessed where applicable):

- Inputs:
    - pass_id in {0,1}
    - pass_assoc in {0,1}
    - pass_nat_or_0 in {0,1} (0 if naturality not present)

- norm_score in [0,1]
- horn_fill_rate in [0,1]
- witness_coverage in [0,1]

- Combine:
  - horn_quality := horn_fill_rate * witness_coverage
  - R_estimate_num := w_id*pass_id*
    *+ w_assoc*pass_assoc
    + w_nat*pass_nat_or_0*
    *+ w_norm*norm_score
    + w_h*horn_quality
  - Z_active := sum of weights actually applied (exclude w_nat if naturality absent)
  - R_estimate := R_estimate_num / max(Z_active, eps)

Weights {w_id, w_assoc, w_nat, w_norm, w_h} are fixed in config. R_estimate never overrides typing or witnesses; it is used only for ranking and reporting.

A5. Pseudocode: integrating R_estimate with existing ai logs

The following is notation-level pseudocode (not tied to a language). All file paths, seeds, compressors, and thresholds must be in the config that produces config_hash.

INPUTS

paths:

in_histories_glob := "C:\\Users\\doug\\ai2\\out\\admissible\\H_*.jsonl"

out_scores_csv := "C:\\Users\\doug\\ai2\\out\\scores\\ranks.csv"

out_byhist_dir := "C:\\Users\\doug\\ai2\\out\\scores\\by_history\\"

config (frozen):

w1, w2

```
    w_id, w_assoc, w_nat, w_norm, w_h
    lambda, a1, a2
    kernel_id, alpha, eps
    compressor_id, compressor_level
    prereg_hash, config_hash

FUNCTION load_histories(glob) -> iterator of H_records
  for each file matching glob:
    for each JSONL line:
      parse into H with fields:
        comp2, S, invariants_fingerprint, horns_total, horns_filled,
        witness_coverage, nf_bundle, horn_bundle, desc_base, desc_prior, ...
      yield H

FUNCTION compute_invariant_features(H) -> feats
  pass_id       := H.invariants_fingerprint.pass_id
  pass_assoc    := H.invariants_fingerprint.pass_assoc
  pass_nat_flag := H.invariants_fingerprint.pass_nat_flag   // true/false
  pass_nat_or_0 := 1 if pass_nat_flag == true else 0
  norm_score    := H.invariants_fingerprint.norm_score
  horns_total   := H.horns_total
  horns_filled  := H.horns_filled
  horn_fill_rate := (horns_filled / horns_total) if horns_total > 0 else 0
```

```
  witness_coverage := H.witness_coverage
  return {pass_id, pass_assoc, pass_nat_or_0, norm_score, horn_fill_rate,
witness_coverage, pass_nat_flag}

FUNCTION compute_R_estimate(feats) -> r_est
  // determine active weight sum
  Z_active := w_id + w_assoc + w_norm + w_h
  if feats.pass_nat_flag == true:
    Z_active := Z_active + w_nat
  num := w_id*feats.pass_id
      + w_assoc*feats.pass_assoc
      + (w_nat*feats.pass_nat_or_0 if feats.pass_nat_flag else 0)
      + w_norm*feats.norm_score
      + w_h*(feats.horn_fill_rate * feats.witness_coverage)
  r_est := num / max(Z_active, eps)
  return r_est

FUNCTION compute_MDL_lift(H) -> mdl
  // desc_base and desc_prior are canonical encodings WITHOUT volatile fields
  kb := compress(H.desc_base, compressor_id, compressor_level)  // returns byte
length
  kp := compress(H.desc_prior, compressor_id, compressor_level)
  mdl := kb - kp
  return mdl
```

```
FUNCTION compute_C_final(H, r_est) -> cf
  mdl := compute_MDL_lift(H)
  cf  := a1*r_est + a2*mdl
  return cf

FUNCTION rank_histories()
  rows := []
  for H in load_histories(in_histories_glob):
    feats := compute_invariant_features(H)
    r_est := compute_R_estimate(feats)
    cf    := compute_C_final(H, r_est)
    sc    := w1*H.comp2 + w2*r_est
    rows.append({
      "history_id": H.id,
      "score": sc,
      "comp2": H.comp2,
      "R_estimate": r_est,
      "C_final": cf,
      "S": H.S,
      "pass_id": feats.pass_id,
      "pass_assoc": feats.pass_assoc,
      "pass_nat_or_0": feats.pass_nat_or_0,
```

```
    "norm_score": feats.norm_score,
    "horn_fill_rate": feats.horn_fill_rate,
    "witness_coverage": feats.witness_coverage,
    "config_hash": config_hash,
    "prereg_hash": prereg_hash
  })
  write_json(out_byhist_dir + "H_" + H.id + ".json", rows[-1])

// deterministic tie-breakers
sort rows by:
  (-score),
  (+S),              // lower action first
  (+timestamp_start),  // earlier first
  (+nf_bundle_hash)    // lexicographic
write_csv(out_scores_csv, rows)

MAIN
assert hashes_in_logs_match(prereg_hash, config_hash)
rank_histories()
emit_log_checksum(out_scores_csv)
```

Notes on determinism and safety

- The pipeline above reads only admissible artifacts. It cannot re-run normalization or generate new witnesses.

- If naturality is not present in a context, pass_nat_or_0 = 0 and w_nat is excluded from Z_active; do not silently count it as pass.
- compress(.) must be the exact preregistered compressor/version/level.
- Tie-breakers ensure identical ordering across machines given identical inputs.

A6. Minimal config JSON (example; values are placeholders)


```
{
  "w1": 1.0,
  "w2": 0.30,
  "w_id": 1.0,
  "w_assoc": 1.0,
  "w_nat": 1.0,
  "w_norm": 2.0,
  "w_h": 2.0,
  "lambda": 0.0,
  "a1": 1.0,
  "a2": 0.0,
  "kernel_id": "cosine_invariants_v1",
  "alpha": 0.50,
  "eps": 1e-12,
  "compressor_id": "zstd_vX",
  "compressor_level": 9,
  "prereg_hash": "sha256:...fixed...",
  "config_hash": "sha256:...autofilled..."
```

}

Bottom line. With this ASCII schema, any third party can take your admissible logs, recompute R_estimate, C_final, and score, and reproduce the same ranks and figures—without touching ai1's proofs or normalization.

Appendix B: Reproducibility Kit

This kit makes every run replayable, hash-verifiable, and comparable across machines—without touching ai1's equational core.

B1. Logging spec (immutable, append-only)

B1.1 Per-step log (JSONL)

Path: C:\Users\doug\ai2\out\steps\run_<RUNID>\steps.jsonl
One JSON object per line. Final line = trailer (see B2.4).

Fields (minimal):

- run_id (str): e.g., "2025-09-09T03-26-28Z_ai2_r001"
- history_id (str): "H_000123"
- step_index (int): 0,1,2,…
- utc_timestamp (str, ISO-8601 Z)
- ruleset_id (str): versioned grammar/type rules
- seed_master (hex): 16 or 32 hex bytes (see B3)
- env_fingerprint (obj): { "os":"Win10_22H2", "python":"3.13.0", "cpu":"Ryzen7-5825U" }
- psi_hash (hex): SHA256 of encoded state
- nf_count (int)
- horns_posed (int), horns_filled (int)
- pass_id (0/1), pass_assoc (0/1), pass_nat ("present_pass"|"present_fail"|"absent")
- norm_score (float in [0,1])
- comp2_step (float)

- R_step (float in [0,1], if computed online else null)
- performance_step (float; preregistered metric)
- discovery_flag (0/1; per Appendix A)

B1.2 Per-history bundle (JSONL)

Path: C:\Users\doug\ai2\out\admissible\H_*.jsonl

Fields (per line):

- id (str): "H_000123"
- start_utc, end_utc (ISO-8601 Z)
- input_set_hash (hex)
- nf_bundle (arr of objs):
  [{"type_sig":"A->B","nf_hash":"hex","norm_trace_hash":"hex"}, ...]
- horn_bundle (arr):
  [
  {"horn_id":"k17","boundary_hash":"hex","filler_hash":"hex","cert_hash":"hex","pass":true}, ...]
- invariants_fingerprint (obj):
  { "pass_id":1, "pass_assoc":1, "pass_nat_flag":false, "norm_score":0.97 }
- horns_total (int), horns_filled (int), horn_fill_rate (float), witness_coverage (float)
- comp2 (float), S (float)
- desc_base (base64) — canonical encoding sans volatile fields
- desc_prior (base64) — prior-factored encoding (same schema)
- R_estimate (float in [0,1]) // filled by Pass B
- C_final (float) // filled by Pass B
- score (float) // filled by Pass B

- ruleset_id, seed_master, env_fingerprint
- config_hash, prereg_hash

B1.3 Scores table (CSV)

Path: C:\Users\doug\ai2\out\scores\ranks.csv
Header + rows:
history_id,score,comp2,R_estimate,C_final,S,pass_id,pass_assoc,pass_nat_or_0,norm_score,horn_fill_rate,witness_coverage,config_hash,prereg_hash

---

B2. File formats & integrity

B2.1 JSONL rules

- UTF-8, one JSON object per line.
- Canonical JSON for objects written by Pass B: keys sorted lexicographically, no trailing spaces.
- Floats emit with max 6 decimals (fixed in config).

B2.2 CSV rules

- Comma-separated, RFC4180, header present, " quotes around fields containing commas/quotes.

B2.3 Hash sidecars

- Every file X has a sidecar X.sha256 containing:
  sha256 <HEX> *<BASENAME>

B2.4 JSONL trailer

- Last line of each JSONL file is a trailer object:

{ "_trailer":"sha256", "file_sha256":"<HEX>", "line_count":123456 }

- Verifiers must recompute hash over all lines excluding the trailer, then match file_sha256.

B3. Seed protocols (deterministic PRNG)

B3.1 Master seed derivation (64-bit)

- Concatenate: run_id + "|" + ruleset_id + "|" + hostname
- seed_master := uint64(SHA256(bytes) mod 2^64)

B3.2 Local seeds

- Per history: seed_H := uint64(SHA256(seed_master || "|" || history_id) mod 2^64)
- Per component: derive sub-seeds with fixed tags, e.g.
  seed_enum := u64(SHA256(seed_H || "|enum") mod 2^64)
  seed_horn := u64(SHA256(seed_H || "|horn") mod 2^64)

B3.3 PRNG choices (fixed)

- Python random.Random(seed) for scalar draws (stable MT19937).
- NumPy Generator(PCG64DXSM(seed)) for arrays.
- Record: {"py_random":"MT19937","numpy":"PCG64DXSM"} in env_fingerprint.

Never reseed mid-history; derive once per scope.

B4. Directory layout (Windows)

C:\Users\doug\ai2\

config\

ai2_scores.config.json

prereg_YYYYMMDD.json

out\

steps\run_<RUNID>\steps.jsonl (+ .sha256)

admissible\H_*.jsonl (+ .sha256)

scores\ranks.csv (+ .sha256)

scores\by_history\H_*.json (+ .sha256)

analysis\

te\ // transfer entropy

mdl\ // compression

cluster\ // discovery clustering

ethics\ // ABAB analyses

figs\

logs\

replay*, audits*

B5. Analysis scripts outline (notation; names and I/O fixed)

1. mdl_compute.py
    - In: admissible H_*.jsonl
    - Args: --compressor zstd_v1.5.6 --level 9
    - Out: analysis\mdl\mdl_results.csv with history_id,K_base,K_prior,Delta_K
2. te_neg_lag.py
    - In: steps steps.jsonl (with discovery flags)
    - Args: --lags -3,-2,-1 --estimator knn --k 4 --window 200

- Out: analysis\te\te_windows.csv and summary te_summary.csv

3. cluster_scan.py
    - In: discovery timestamps from steps/admissible
    - Args: --windows 30min,60min,120min --bootstrap 10000
    - Out: analysis\cluster\scan_stats.csv with p-values
4. ethics_abab.py
    - In: block schedule analysis\ethics\blocks.csv, ranks/steps
    - Args: --model mixed --alpha 0.01
    - Out: analysis\ethics\effects.csv (effect sizes, CIs)
5. make_figures.py
    - In: all above CSVs
    - Out: analysis\figs\fig_*.png (TE plots, MDL violin, cluster scan, ABAB bars)
6. replay_verify.ps1 (PowerShell)
    - Function: re-run hash checks, JSONL trailer validation, config/prereg hash match, and determinism replay on a sample of histories.
    - Out: logs\replay\report_<RUNID>.txt

Run order (example):

mdl_compute.py

te_neg_lag.py

cluster_scan.py

ethics_abab.py

make_figures.py

replay_verify.ps1

B6. Canonical encoding for MDL (no leakage)

- desc_base and desc_prior are byte-identical across replays.
- Schema excludes volatile fields: timestamps, hashes, scores.
- Prior factoring moves repeated invariant tokens into a reference table; it cannot embed computed summaries (e.g., “final answer”).
- The schema version string is in config_hash; compressors recorded as {"codec":"zstd","version":"1.5.6","level":9}.

B7. Replay recipe (operator checklist)

1. Confirm config_hash and prereg_hash printed at run start.
2. After run, verify sidecars: *.sha256.
3. Validate trailers on steps.jsonl and H_*.jsonl.
4. Run analysis scripts in B5 (no flags changed).
5. Execute replay_verify.ps1 (rebuilds a sample; byte-diff proofs/logs).
6. Archive: zip out\ and analysis\ with SHA256 and store DOI or RG artifact link.

B8. Rotation & archival

- Chunking: split steps.jsonl daily or every 5M lines.
- Compression: .jsonl.zst allowed after sidecar/trailer creation; produce new sidecar for the compressed file.
- Retention: raw + compressed kept 1 year local; compressed pushed to cold storage with manifest:
    - manifest.json: list of files with size, sha256, creation UTC, run_id.

B9. Common failure modes & quick triage

- Trailer hash mismatch: file truncated or edited → recover from latest zipped copy; never “fix in place.”
- Determinism drift: check ruleset_id or PRNG seeds; ensure no unsorted parallel outputs.
- Pass-by-absence inflation: in scores, pass_nat_or_0 must be 0 when pass_nat_flag=false.
- Compressor mismatch: MDL outliers → confirm codec ver/version/level matches config.

B10. Minimal example rows (illustrative)

Scores CSV row:

H_000123,1.284,1.011,0.422,0.422,0.337,1,1,0,0.97,0.88,0.92,sha256:a1..,sha256:b2..

JSONL trailer:

{ "_trailer":"sha256", "file_sha256":"D4A7...9F", "line_count": 1048577 }

Bottom line. Fixed schemas, deterministic seeds, hashed sidecars, JSONL trailers, and a frozen analysis toolchain ensure that anyone—now or years later—can verify artifacts, recompute scores, and reproduce figures bit-for-bit, while the proof kernel remains untouched.

Figure 1. Resonant Epistemology: Logos, Retrocausality, and Discovery-Only AI