**Deterministic Witnesses, Generative Drafts: The ChatGPT × youvan.ai Hybrid Research Method**

Douglas C. Youvan

doug@youvan.com

www.youvan.ai

September 17, 2025

This paper formalizes the working method that has emerged from our recent collaborations: a hybrid research lab where Douglas C. Youvan (human) and ChatGPT (AI) operate as one coordinated system. The human sets aims, priors, and adjudication criteria; the AI drafts, synthesizes, and scaffolds code, figures, and art. We keep a single narrative voice, but every artifact carries dual provenance. Deterministic components—copy-safe math, schemas, scripts, and data—ship with witnesses, hashes, and replay instructions. Generative components—prose, diagrams, and exploratory prompts—are labeled with model family and run metadata. The result is a repeatable pipeline: ideas become artifacts; artifacts become indexed, reusable assets; reuse compounds into faster, clearer science. We document the production loop (Prepare → Refresh → Generate → Verify → Publish → Index → Iterate), the role split (aim selection vs. expansion), and the reproducibility kit (prompt seeds, SHA256, content addressing, minimal counter-witnesses). We propose metrics for progress (reuse ratio, time-to-half-output, proof density) and governance guardrails (speculation labels, refusal tests, after-action reviews). Case vignettes show the method in CT/HoTT-flavored symbolic AI and biophysics writing. Limitations and falsifiers are explicit: where witnesses are thin, claims are modest; where reuse fails, we revise or retire. The north star is public benefit: clarity, peace, and shareable, verifiable work.

Keywords: ChatGPT × youvan.ai, hybrid research, centaur intelligence, proof-carrying artifacts, copy-safe math, reproducibility, witnesses, SHA256, content addressing, prompt provenance, CT/HoTT, ai1 method, GenesisBoard, research pipeline, governance, metrics, JSON-LD index, seed registry.

A collaboration with GPT-5. CC4.0. 59 pages.

**Outline**

1. **Thesis: Why Hybridize Now**
   1.1 The single-voice, dual-provenance principle
   1.2 From gradient lore to proof-carrying reuse
   1.3 Scope: symbolic AI, biophysics, pedagogy
2. **Roles and Responsibilities**
   2.1 Human: aims, priors, adjudication, ethics
   2.2 AI: drafting, synthesis, scaffolds (code/art)
   2.3 Hand-off protocol and “one-voice” editing
3. **Production Loop**
   3.1 Prepare (aims, priors, sources)
   3.2 Refresh (seeds, registries, prior artifacts)
   3.3 Generate (prose, code, figures, prompts)
   3.4 Verify (witnesses, hashes, counter-witnesses)
   3.5 Publish (paper + kit), Index (JSON-LD), Iterate
4. **Deterministic Core**
   4.1 Copy-safe math and executable schemas
   4.2 Content addressing and SHA256 fingerprints
   4.3 Minimal counter-witness format and replay protocol
5. **Generative Layer**
   5.1 Prompt provenance: model, date, seed sketch
   5.2 Diagrams and lead images: labeled as generative
   5.3 Boundaries: where generative must yield to proof
6. **Reproducibility Kit**
   6.1 File and folder conventions; seed registry
   6.2 Logs, run cards, and artifact manifests
   6.3 JSON-LD research index and cross-paper linking
7. **Metrics and KPIs**
   7.1 Reuse ratio (RR), proof density (PD)
   7.2 t_half_out and path-entropy proxy
   7.3 Repair/restore rates; auditability score
8. **Governance and Ethics**
   8.1 Speculation labels; refusal tests; after-action review

## 1. Thesis: Why Hybridize Now

### 1.1 The single-voice, dual-provenance principle

**Claim.** Readers deserve one clear narrative voice; auditors deserve two clear sources. We therefore write as a single stylistic voice while attaching **dual provenance** to every artifact: *(human: Douglas C. Youvan; AI: ChatGPT — model + date + run notes).*
**Why now.** The volume and tempo of hybrid work make ad-hoc acknowledgments insufficient. Without a principled split, authorship blurs, reproducibility stalls, and ethics drift.
**Operational rules.**

- **Voice:** one editorial tone throughout; no alternating “human says / AI says” blocks except in methods notes.
- **Attribution tags:** each generative segment carries (Model=GPT-5 Thinking, Date=YYYY-MM-DD, Seed=<sketch>).

- **Deterministic artifacts:** copy-safe math, code, schemas, and data ship with witnesses: hash=SHA256(file), replay=script.
- **Edits & decisions:** human adjudication is explicit at section ends ("Human final pass: …").
  **Edge cases.** Direct quotations, citations, and legally sensitive claims retain their own provenance; AI-assisted phrasing remains in the single voice but is labeled as generative. This balances readability with auditability.

### 1.2 From gradient lore to proof-carrying reuse

**Problem.** Modern AI practice often relies on "gradient lore": undocumented prompt tricks, brittle settings, and unverifiable hunches. Outcomes are hard to audit, harder to reuse.
**Shift.** We prioritize **proof-carrying reuse**: artifacts that bring their own verification and integration keys. Examples include:

- **Typed rewrites to NF:** e.g., g ∘ id_B ∘ f → g ∘ f, pushout horn checks, and mid-identity elimination recorded as machine-checkable steps.
- **Witness bundles:** {artifact, SHA256, run_card.json, counter_witness_min.json} enabling third-party replay or challenge.
- **Content addressing:** references point to hashes, not filenames; reuse = "pull by hash," not "copy by hunch."
  **Payoff.** Reuse compounds: each verified unit becomes a building block. Metrics shift from raw FLOPs to **RR (reuse ratio)**, **PD (proof density)**, and **t_half_out (time to half the final output quality)**. Where witnesses thin out—e.g., speculative diagrams or exploratory prose—we label them, bound their scope, and invite falsifiers. The goal is to replace mystique with a **supply chain of proofs** that accelerates future work.

### 1.3 Scope: symbolic AI, biophysics, pedagogy

**Symbolic AI.** Primary testbed. We target equational engines, category-theoretic schemas, HoTT-style normal forms, and JSON-LD research indices. Deterministic cores (rewrites, checks, hashes) pair with labeled generative scaffolds (explanations, figures, prompts).
**Biophysics.** Focus on *explanatory modeling* and literature-synthesis, not clinical guidance. Copy-safe derivations (e.g., dimensional analyses, simple kinetics) are supplied with calculation logs and data hashes; speculative mechanisms are clearly marked and bounded by falsifiers and references.
**Pedagogy.** CT/HoTT teaching tools (Twin Cubes, cube nets, proof cards) are ideal for the hybrid loop: the AI drafts assets and variations; the human curates, simplifies, and standardizes. Classroom-ready kits ship with print templates, run cards, and license terms.
**Out-of-scope (for now).** Wet-lab protocols, undisclosed patient data, and high-stakes medical or

financial advice. Where empirical claims require external datasets, we either include consented, hashed samples or treat results as provisional.
**North star.** Public-benefit artifacts that are readable on first pass, auditable on second, and reusable on third—linking symbolic AI, biophysics exposition, and pedagogy under one hybrid method.

## 2. Roles and Responsibilities

### 2.1 Human: aims, priors, adjudication, ethics

**Aims.** Choose problems, define success/falsifiers, set scope and audience.
**Priors.** Provide domain constraints (definitions, lemmas, references), preferred formalisms, and house style (Copy-Safe ASCII Math; section architecture; figure conventions).
**Adjudication.** Final decision rights on: (i) claims admitted; (ii) what moves to appendix; (iii) when to retire or revise. Maintain a lightweight **Decision Log**: date • section • keep/strike • rationale.
**Source discipline.** Supply canonical sources; flag unstable facts for web verification; require citations where claims could have changed since publication.
**Witness gate.** Demand replayable artifacts (hashes, run cards) for deterministic results; insist that speculative visuals/text be labeled and bounded.
**Ethics.** Public-benefit posture, no deification, clear speculation labels, license hygiene (text CC BY 4.0; code MIT/Apache-2.0), conflict disclosures, and refusal tests for safety.
**Editorial duty.** Unify tone, prune redundancy, and enforce “one-voice” readability.

**Human acceptance checklist (per section):** scope fit ✓ | claims testable or labeled ✓ | copy-safe math ✓ | citations/witnesses ✓ | plain-English summary ✓.

---

### 2.2 AI: drafting, synthesis, scaffolds (code/art)

**Drafting.** Expand outlines into prose; tighten arguments; surface alternatives; compress jargon; produce executive summaries.
**Synthesis.** Consolidate cross-paper motifs; align terminology; reconcile notations; generate comparison tables and glossaries.
**Scaffolds.**

- **Code/data:** minimal reproducible examples, schema files, validators, and run cards (model, date, seed_sketch, inputs, outputs, SHA256).

- **Figures/art:** promptable diagrams and lead images, each tagged with model/date and marked “generative.”
- **Indexing:** JSON-LD entries, citation stubs, artifact manifests, and content-addressed links.
  **Verification assists.** Compute hashes; run deterministic checks; draft counter-witness sketches (how a claim could fail); flag unstable facts for browsing and attach citations when used.
  **Guardrails.** No fabricated references; uncertain items are explicitly marked “uncertain”; default to minimal claims when witnesses are thin; respect safety and licensing boundaries.

**AI handoff checklist:** outline coverage ✓ | facts stable or cited ✓ | witnesses/hashes present ✓ | speculation labeled ✓ | style hints inserted for one-voice edit ✓.

---

**2.3 Hand-off protocol and “one-voice” editing**

**Artifacts & names.**

- paper_draft_vN.md (single voice, section stamps)
- kit/ → run_card.json, artifacts.json, manifest.json, SHA256SUMS.txt
- fig/ → generative sources + captions + tags
- index/entry.jsonld (for cross-paper linking)

**Pass ordering.**

1. **Generate (AI)** → complete draft + kit skeleton.
2. **Verify (AI)** → hashes, counters, unstable-facts flags.
3. **Adjudicate (Human)** → keep/strike decisions in Decision Log.
4. **Unify (Human)** → one-voice edit: merge diction, tighten claims, normalize notation, ensure Copy-Safe ASCII Math.
5. **Stamp (Both)** → insert **Hybrid Note**, model/date stamps, licenses, and final SHA256 file.

**One-voice rules.** No “human/AI” alternation in body; use a single narrative register; keep provenance in headers/footers and figure notes. Prefer short, active sentences; define terms once; push heavy detail to appendices with stable anchors.

**Change discipline.** All edits via tracked diffs; commit messages in the form date • section • verb • reason (e.g., 2025-09-17 • 4.2 • tighten • removed duplicate lemma, added hash).

**Stop conditions.** Publish when: witnesses complete ✓ | unstable facts either cited or removed ✓ | Decision Log closed ✓ | readability pass cleared by a third-party skim test ✓.

**You said:**

Expand: Production Loop 3.1 Prepare (aims, priors, sources) 3.2 Refresh (seeds, registries, prior artifacts) 3.3 Generate (prose, code, figures, prompts) 3.4 Verify (witnesses, hashes, counter-witnesses) 3.5 Publish (paper + kit), Index (JSON-LD), Iterate

**3. Production Loop**

**3.1 Prepare (aims, priors, sources)**

**Goal.** Enter with a falsifiable aim, bounded scope, and a short list of canonical sources. Exit with a ready-to-run folder skeleton.

**Inputs.**

- **Aim statement (1–3 sentences).** Problem, success criteria, explicit falsifiers, audience.
- **Priors.** Definitions, lemmas, notations, house style (Copy-Safe ASCII Math), and any constraints.
- **Sources plan.** Canonical refs to consult; facts likely to have changed flagged for verification; prior internal artifacts (by hash) to reuse.

**Artifacts.**

- prep/aims.md (aim, success, falsifiers, audience)
- prep/priors.md (terms, symbols, style)
- prep/sources.md (citations to check; “unstable facts” list)
- Project scaffold:
- paper_draft_vN.md
- kit/run_card.json
- kit/manifest.json
- fig/ (empty)

- index/entry.jsonld

**Checklist.** scope fit ✓ | falsifiers present ✓ | priors aligned ✓ | sources identified ✓ | scaffold built ✓

---

**3.2 Refresh (seeds, registries, prior artifacts)**

**Goal.** Pull forward what works—reseed from proven prompts and pull artifacts by hash.

**Inputs.**

- **Seed set.** 3–7 short “seed sketches” naming tone, constraints, target structure.
- **Registries.** Local registry of seeds and artifacts; cross-paper JSON-LD index.
- **Prior artifacts.** Code, schemas, figures previously verified (referenced by SHA256).

**Artifacts.**

- kit/seed_registry.json (minimal example):
- {
- "series": "ai1-hybrid",
- "seeds": [
- {"id":"20250917-hybrid-core","sketch":"single-voice, dual-provenance; witnesses-first"},
- {"id":"20250917-det-core","sketch":"typed rewrites; NF; counter-witness"}
- ]
- }
- kit/artifacts.json (content-addressed):
- {
- "artifacts":[
- {"name":"rewrite_engine_min.py","sha256":"<hash>"},
- {"name":"copy_safe_math_guide.txt","sha256":"<hash>"}
- ]

- }

**Checklist.** seeds selected ✓ | prior hashes resolved ✓ | compatibility notes captured ✓

---

**3.3 Generate (prose, code, figures, prompts)**

**Goal.** Draft everything in one pass, but keep deterministic and generative layers cleanly separated.

**Prose.**

- Expand outline → full sections.
- Keep a single register; leave [HUMAN_PASS_NOTE: …] markers where adjudication is needed.
- Short paragraph caps per section for readability.

**Code/data.**

- Minimal reproducible examples only.
- Include validators and typed rewrite logs.
- Small, synthetic datasets with generation scripts.

**Figures/art.**

- Diagrams and lead images labeled **generative** with model+date.
- Captions contain the claim the figure supports (or say “illustrative only”).

**Prompts.**

- Store prompt scaffolds with seed IDs, not raw long prompts inside the paper body.

**Artifacts.**

- paper_draft_vN.md (complete first pass)
- kit/run_card.json:
- {
- "model":"GPT-5 Thinking",
- "date_local":"2025-09-17",

- "timezone":"America/Chicago",
- "seeds":["20250917-hybrid-core","20250917-det-core"],
- "inputs":["prep/aims.md","prep/priors.md","prep/sources.md"],
- "outputs":["paper_draft_vN.md","fig/lead.png","kit/manifest.json"]
- }
- fig/lead_src.txt (prompt sketch + tags)

**Checklist.** outline fully covered ✓ | code runs locally ✓ | figures tagged ✓ | prompts externalized ✓

---

**3.4 Verify (witnesses, hashes, counter-witnesses)**

**Goal.** Turn claims into auditable units; compute hashes; write down how the claim could fail.

**Witness types.**

- **Deterministic math/code.** Typed rewrite logs, unit tests, dimension checks.
- **Data/figures.** Source description, generation script, parameters.
- **Text claims.** Citation tie-back or "speculative" label.

**Counter-witness (minimal).** For each nontrivial claim, add a one-liner: "This would be false if …" plus a test or reference likely to refute.

**Hashes & manifests.**

- Compute SHA256 for all non-ephemeral files.
- Build a single manifest and sum file.

**Artifacts.**

- kit/SHA256SUMS.txt
- kit/manifest.json (map logical names → sha256)
- kit/counter_witness_min.json:
- {
- "claims":[

- {"id":"C1","text":"Proof-carrying reuse reduces time-to-half-output.",
- "would_fail_if":"Reuse ratio stays flat across 3 papers.",
- "test_plan":"Track RR over next 3 publications; require RR>=0.25"}
- ]
- }

**Checklist.** all files hashed ✓ | witnesses attached ✓ | counter-witness per claim ✓ | unstable facts cited or removed ✓

---

**3.5 Publish (paper + kit), Index (JSON-LD), Iterate**

**Goal.** Ship a cohesive paper + reproducibility kit; cross-link it; learn from reuse.

**Publish.**

- **Stamp.** Insert Hybrid Note (model/date), licenses (CC BY 4.0 text; MIT/Apache-2.0 code).
- **Package.** Paper PDF/MD + kit/ + fig/ + SHA256SUMS.txt.
- **Release target.** ResearchGate DOI or repository; include CITATION.cff.

**Index (JSON-LD).**

- Cross-paper linking and discoverability via a minimal entry.

index/entry.jsonld (skeleton):

```
{
 "@context":"https://schema.org",
 "@type":"CreativeWork",
 "name":"Deterministic Witnesses, Generative Drafts",
 "author":[
  {"@type":"Person","name":"Douglas C. Youvan"},
  {"@type":"Organization","name":"ChatGPT","alternateName":"GPT-5 Thinking"}
 ],
 "datePublished":"2025-09-17",
```

"keywords":["hybrid research","witnesses","copy-safe math","CT/HoTT"],
"isPartOf":{"@type":"CreativeWorkSeries","name":"ai1-hybrid"},
"identifier":[
{"@type":"PropertyValue","propertyID":"sha256","value":"<top-level-bundle-hash>"},
{"@type":"PropertyValue","propertyID":"doi","value":"<doi-or-rg-link>"}
],
"hasPart":[
{"@type":"MediaObject","name":"paper_draft_vN.md","contentUrl":"./paper_draft_vN.md"},
{"@type":"MediaObject","name":"kit/manifest.json","sha256":"<hash>"}
]
}

**Iterate.**

- **Post-mortem (24–72h).** What was reused? What broke? Update counter-witness outcomes.
- **Metrics.**
  - RR = reused_units / total_units (target: monotone non-decreasing across papers)
  - PD = proofs_or_witnesses / material_claims (target: >= 0.7 outside clearly labeled speculation)
  - t_half_out = elapsed time until 50% of final sections stabilize (target: declines over series)
- **Ledger updates.** Append to kit/run_card.json (“reuse_log”, “fix_log”). Promote stable pieces to the seed registry.

**Ship criteria.** witnesses complete ✓ | manifest & sums present ✓ | JSON-LD valid ✓ | Hybrid Note & licenses present ✓ | skim test passed ✓

**Rinse → repeat.** Each cycle should lower t_half_out, raise RR, and keep PD high. If any metric regresses twice in a row, schedule a “methods tune-up” before the next paper.

## 4. Deterministic Core

### 4.1 Copy-safe math and executable schemas

**House style (Copy-Safe ASCII Math).**

- Composition: g \circ f, identity: id_A, inverse: u^{-1}, product: A \times B, sum: A + B, hom: Hom(A,B).
- Rewrites are written as LHS -> RHS with a named rule, e.g., g \circ id_B \circ f -> g \circ f (mid-identity elimination).
- Typing is explicit when helpful: f: A -> B, g: B -> C.

**Executable rewrite log (schema sketch).**
Each deterministic derivation ships as a machine-checkable JSON log.

```
{
  "proof_id": "C1",
  "goal": "g \\circ id_B \\circ f -> g \\circ f",
  "context": {"f":"A->B","g":"B->C","id_B":"B->B"},
  "steps": [
    {
      "lhs":"g \\circ id_B \\circ f",
      "rule":"mid_identity_elim",
      "rhs":"g \\circ f",
      "check":"type_safe"
    }
  ],
  "result":"proved",
  "engine_sha256":"<sha256-of-validator>"
}
```

**Validator contract (minimal).**

- Input: rewrite_log.json (as above) + library of rules (rules.json).
- Must verify: (i) type safety per step, (ii) rule applicability, (iii) LHS/RHS normalization equivalence where relevant.
- Output: rewrite_log.report.json with {proved|failed}, and a per-step verdict.

**Executable data schemas (tiny).**

- rules.json entry:

{"name":"mid_identity_elim","pattern":"g \\circ id_B \\circ f","rewrite":"g \\circ f","side_conditions":["f: A->B","g: B->C"]}

---

### 4.2 Content addressing and SHA256 fingerprints

**Principle.** Every artifact is addressed by its content, not its filename. We use sha256:<64hex> as the canonical address.

**Manifests.**

- kit/manifest.json (logical name → sha256):

{
  "artifacts": [
    {"name":"paper_draft_v3.md","sha256":"<hash>"},
    {"name":"rewrite_engine_min.py","sha256":"<hash>"},
    {"name":"rewrite_log.json","sha256":"<hash>"}
  ]
}

- kit/SHA256SUMS.txt (portable):

<hash>  paper_draft_v3.md
<hash>  rewrite_engine_min.py
<hash>  rewrite_log.json

**Top-level bundle hash (deterministic).**
Compute a **bundle hash** over a canonical JSON that includes sorted artifact entries and their hashes; newline = \n, UTF-8, sorted keys. This avoids toolchain differences (tar/zip).

**How to compute (cross-platform).**

- GNU/Unix: sha256sum file.ext
- PowerShell (Windows):

```
$start = Get-Date

Get-FileHash -Algorithm SHA256 .\paper_draft_v3.md | Format-List

Get-FileHash -Algorithm SHA256 .\rewrite_engine_min.py | Format-List

Get-FileHash -Algorithm SHA256 .\rewrite_log.json | Format-List

$end = Get-Date; "Start: $start  End: $end"
```

**Linking by hash.**
In prose and kits, prefer sha256:<hash> references. Filenames are convenience only. When reusing, **pull by hash**, not by path.

---

**4.3 Minimal counter-witness format and replay protocol**

**Why.** Every nontrivial claim should ship with how it could fail and how to test that.

**Counter-witness (minimal JSON).**

```
{
  "claims":[
    {
      "id":"C1",
      "text":"g \\circ id_B \\circ f -> g \\circ f holds under stated types.",
      "would_fail_if":"Type discipline is violated or rule mid_identity_elim is mis-specified.",
      "test_plan":"Run validator on rewrite_log.json using rules.json; expect result=proved.",
      "status":"planned",  // planned | running | sustained | falsified
      "evidence":[
```

{"sha256":"<rewrite_log.json hash>"},
{"sha256":"<rules.json hash>"}
],
"last_checked":"2025-09-17"
}
]
}

**Replay protocol (deterministic, offline).**

1. **Fetch by hash.** Obtain required artifacts by sha256 from the manifest.
2. **Verify integrity.** Check each file against kit/SHA256SUMS.txt.
3. **Run validator.**
    - Python example: python validate.py --log rewrite_log.json --rules rules.json --out rewrite_log.report.json
4. **Compare expected vs observed.**
    - Expected: result="proved" (or specific normal form).
    - If mismatch, mark claim falsified and emit diff.
5. **Record replay.**
    - Append to kit/replay_report.json:

{
"claim":"C1",
"replayer":"third_party_or_self",
"timestamp":"2025-09-17T16:10:00-05:00",
"inputs":[
{"name":"rewrite_log.json","sha256":"<hash>"},
{"name":"rules.json","sha256":"<hash>"}
],

```
  "observed_result":"proved",
  "tooling":{"validator_sha256":"<hash>"}
}
```

**One-click scripts (both shells).**

- kit/replay.sh (POSIX):

```
#!/usr/bin/env sh
set -eu
echo "Start: $(date -Is)"
sha256sum -c kit/SHA256SUMS.txt
python validate.py --log rewrite_log.json --rules rules.json --out rewrite_log.report.json
echo "OK: validator completed"
echo "End: $(date -Is)"
```

- kit/replay.ps1 (Windows, with timestamps):

```
$start = Get-Date; "Start: $start"
Get-Content .\kit\SHA256SUMS.txt | ForEach-Object {
  $parts = $_ -split '\s+'
  if ($parts.Length -ge 2) {
    $expected = $parts[0]; $file = $parts[-1]
    $actual = (Get-FileHash -Algorithm SHA256 $file).Hash.ToLower()
    if ($actual -ne $expected) { throw "Hash mismatch: $file" }
  }
}
python .\validate.py --log .\rewrite_log.json --rules .\rules.json --out .\rewrite_log.report.json
"OK: validator completed"
$end = Get-Date; "End: $end"
```

**Exit criteria (per claim).** status = sustained only if: hashes match ✓, validator passes ✓, replay report written ✓. Otherwise mark falsified and open a revision ticket.

## 5. Generative Layer

### 5.1 Prompt provenance: model, date, seed sketch

**Principle.** Keep the paper in one human-readable voice, but tag any AI-generated assistance with compact, auditable provenance.

**What to record (minimal).**

- model: e.g., GPT-5 Thinking
- date_local + timezone: e.g., 2025-09-17, America/Chicago
- seed_id: short ID tying to the seed registry
- seed_sketch: 1–2 lines describing intent/constraints (not the full prompt)
- purpose: prose | diagram_prompt | code_scaffold | glossary | summary
- inputs: hashes of any files the AI was shown
- output_refs: hashes of produced artifacts

**Where to store.** In the kit, not the body text:

- kit/prompt_provenance.jsonl (one JSON per generative call)
- Paper body includes only a tiny footnote or the **Hybrid Note** pointing to this file.

**Example (one line in prompt_provenance.jsonl):**


{

"model":"GPT-5 Thinking",

"date_local":"2025-09-17",

"timezone":"America/Chicago",

"seed_id":"20250917-hybrid-core",

"seed_sketch":"single-voice, dual-provenance; witnesses-first",

"purpose":"prose",

"inputs_sha256":["<aims.md#sha>", "<priors.md#sha>"],

"output_sha256":["<paper_draft_v3.md#sha>"]

}

**Seed registry (separate).**

- kit/seed_registry.json keeps 3–7 reusable seeds with terse sketches.
- Full prompts (if needed) live in a private appendix or external repo; the paper never embeds long prompts.

---

**5.2 Diagrams and lead images: labeled as generative**

**Two classes.**

1. **Illustrative generative art** (cover/lead images, concept sketches).
    - Always labeled "generative"; never presented as empirical evidence.
    - Caption begins with Illustrative. and states the idea conveyed, not a result.
2. **Evidence-bearing figures** (plots, tables, rewrites).
    - **Not** generative art. Must be produced by code from data or logs, with hashes for code, data, and figure files.

**Folder & tags.**

- fig/lead_src.txt → prompt sketch + tags + model/date
- fig/lead.png → the rendered image (with SHA256)
- fig/diagram_* for concept diagrams; fig/plot_* reserved for evidence

**Caption patterns.**

- *Generative (illustrative):*
  "**Illustrative.** Conceptual map of a single-voice / dual-provenance workflow (generative image, GPT-5 Thinking, 2025-09-17)."
- *Deterministic (evidence):*
  "**Result.** RR (reuse ratio) across papers 1–3; computed by metrics.py from reuse_log.json (sha256: …)."

**Accessibility & licensing.**

- Every image gets alt text in fig/alt.json.
- Default license: images CC BY 4.0 unless external assets force another license (record in fig/LICENSE.txt).

**Rule of separation.**

- If a figure influences a claim, it must be reproducible from code/data.
- Generative art does **not** support claims—only comprehension.

---

**5.3 Boundaries: where generative must yield to proof**

**Red lines (generative → deterministic).**

1. **Mathematical claims.** Any statement that can be reduced to typed rewrites, derivations, or checks must ship with a rewrite log and validator report.
2. **Empirical statements.** Any assertion about measured values, benchmarks, or data trends requires data + code + hashes; no generative substitutes.
3. **Safety/ethics-sensitive content.** Biomedical, legal, or financial guidance demands citations, date-verified facts, and explicit limitations—or it is removed.
4. **Attribution & quotations.** Direct quotes and specific facts that may change over time require sources and dates; otherwise, downgrade to paraphrase or drop.
5. **Diagrams that imply evidence.** If a diagram could be mistaken for data, it must be emitted by code from inputs with hashes; else mark "Illustrative."

**Decision prompts (apply before publication).**

- *Could this be computed, checked, or replayed?* If yes → produce witnesses.
- *Would a skeptical reader need to run this?* If yes → add code, logs, and SHA256.
- *Is any part unstable over time?* If yes → cite and date-stamp, or excise.

**Downgrade path.**

- When proof/witness work is infeasible in time: move the material to an appendix labeled **Speculative** with a counter-witness entry stating how it could fail and what test would decide it.

**Upgrade path.**

- When a previously generative explanation becomes formalizable: replace with deterministic artifact, update the manifest, and close the counter-witness with a replay report.

**Publication gate (Generative Layer).**

- Generative items are present **only** to aid comprehension, are visibly labeled, have provenance in prompt_provenance.jsonl, and make **no** unsupported claims.
- Anything that *does* support a claim lives in the Deterministic Core with code, logs, and hashes.

## 6. Reproducibility Kit

### 6.1 File and folder conventions; seed registry

**Purpose.** A predictable kit lets third parties rebuild, replay, and diff your work without guesswork.

**Folder layout (relative paths).**

```
/            # project root
paper_draft_vN.md
LICENSE.txt     # text: CC BY 4.0; code: MIT/Apache-2.0 per file notes
CITATION.cff
prep/          # aims, priors, sources
  aims.md
  priors.md
  sources.md
fig/           # images & diagrams
  lead.png
  lead_src.txt  # prompt sketch + tags + model/date
  alt.json      # alt text map
```

```
kit/          # everything needed to REPLAY
  run_card.json
  manifest.json
  SHA256SUMS.txt
  counter_witness_min.json
  prompt_provenance.jsonl
  seed_registry.json
  replay.sh
  replay.ps1
  validate.py   # minimal validator if applicable
index/
  entry.jsonld  # JSON-LD record for this paper
logs/         # human/system logs (optional)
  decision_log.md
```

**Naming conventions.**

- Use snake_case for files; prefer ASCII-only names.
- Stamp versions with vN and dates with YYYYMMDD (local time = America/Chicago).
- Never put secrets or long raw prompts in the repo; keep **seed sketches** only.

**Seed registry (kit/seed_registry.json).**

- Keep 3–7 reusable seeds; short descriptions; lifecycle states.

```
{
  "series":"ai1-hybrid",
  "updated":"2025-09-17",
  "seeds":[
    {"id":"20250917-hybrid-core","sketch":"single-voice; dual-provenance; witnesses-first","state":"active"},
```

{"id":"20250917-det-core","sketch":"typed rewrites; NF; counter-witness","state":"active"},
{"id":"20250830-expo-lite","sketch":"short executive summaries; no jargon","state":"draft"}
]
}

**Seed lifecycle.** draft → active → retired. Retire seeds that inflate edits or lower reuse ratio (RR). Promote only after two papers show gains in t_half_out or PD.

---

**6.2 Logs, run cards, and artifact manifests**

**Run card (kit/run_card.json).** One file to tell *what ran, where, with what*.

{
"title":"Deterministic Witnesses, Generative Drafts",
"series":"ai1-hybrid",
"model":"GPT-5 Thinking",
"date_local":"2025-09-17",
"timezone":"America/Chicago",
"environment":{
"os":"Windows 11 / Ubuntu 24.04 (replayer choice)",
"python":"3.13.x",
"tools":{"openssl":"3.x","powershell":"7.x"}
},
"seeds":["20250917-hybrid-core","20250917-det-core"],
"inputs":[
{"name":"prep/aims.md","sha256":"<hash>"},
{"name":"prep/priors.md","sha256":"<hash>"},
{"name":"prep/sources.md","sha256":"<hash>"}
],

"outputs":[
{"name":"paper_draft_v3.md","sha256":"<hash>"},
{"name":"fig/lead.png","sha256":"<hash>"}
],
"metrics":{"RR":null,"PD":null,"t_half_out":null}, // filled post-pub
"notes":"Single-voice edit completed; unstable facts cited or removed."
}

**Artifact manifest (kit/manifest.json).** Canonical map of logical names → SHA256. Keys sorted; LF newlines.

{
"artifacts":[
{"name":"paper_draft_v3.md","sha256":"<hash>"},
{"name":"kit/run_card.json","sha256":"<hash>"},
{"name":"kit/counter_witness_min.json","sha256":"<hash>"},
{"name":"fig/lead.png","sha256":"<hash>"}
],
"bundle_sha256":"<hash-of-this-sorted-json>"
}

**SHA sums (kit/SHA256SUMS.txt).**

<hash> paper_draft_v3.md
<hash> kit/run_card.json
<hash> kit/counter_witness_min.json
<hash> fig/lead.png

**Prompt provenance (kit/prompt_provenance.jsonl).** One JSON per generative assist.

{"model":"GPT-5 Thinking","date_local":"2025-09-17","timezone":"America/Chicago","seed_id":"20250917-hybrid-core","seed_sketch":"single-voice; dual-provenance; witnesses-

first","purpose":"prose","inputs_sha256":["<aims#sha>","<priors#sha>"],"output_sha256":["<paper#sha>"]}

**Counter-witnesses (kit/counter_witness_min.json).** Minimal falsifier entries per claim (see §4.3).

**Decision log (logs/decision_log.md).** Lightweight human adjudication trail.

2025-09-17 • §4.2 • keep • content-addressing needed for replay.

2025-09-17 • §5.2 • tighten • caption clarifies “illustrative”.

**Packing & verification scripts.**

- POSIX kit/replay.sh:

```
#!/usr/bin/env sh
set -eu
echo "Start: $(date -Is)"
sha256sum -c kit/SHA256SUMS.txt
python validate.py --log rewrite_log.json --rules rules.json --out rewrite_log.report.json || true
echo "End: $(date -Is)"
```

- PowerShell kit/replay.ps1:

```
$start=Get-Date; "Start: $start"
Get-Content .\kit\SHA256SUMS.txt | ForEach-Object {
  $p = $_ -split '\s+'
  if ($p.Length -ge 2) {
    $expected=$p[0]; $file=$p[-1]
    $actual=(Get-FileHash -Algorithm SHA256 $file).Hash.ToLower()
    if ($actual -ne $expected) { throw "Hash mismatch: $file" }
  }
}
if (Test-Path .\validate.py) {
  python .\validate.py --log .\rewrite_log.json --rules .\rules.json --out .\rewrite_log.report.json
```

```
}
$end=Get-Date; "End: $end"
```

**CITATION file (CITATION.cff, excerpt).**

```
cff-version: 1.2.0
title: "Deterministic Witnesses, Generative Drafts"
authors:
  - family-names: Youvan
    given-names: Douglas C.
  - name: ChatGPT (GPT-5 Thinking)
date-released: 2025-09-17
doi: "<doi-or-rg-link>"
keywords: [hybrid research, witnesses, CT/HoTT]
```

---

### 6.3 JSON-LD research index and cross-paper linking

**Why JSON-LD.** It's a lightweight, crawlable catalog for papers, kits, and parts—supporting *discoverability*, *cross-linking*, and *content addressing*.

**Per-paper entry (index/entry.jsonld).**

```
{
  "@context":"https://schema.org",
  "@type":"ScholarlyArticle",
  "name":"Deterministic Witnesses, Generative Drafts",
  "author":[
    {"@type":"Person","name":"Douglas C. Youvan"},
    {"@type":"Organization","name":"ChatGPT","alternateName":"GPT-5 Thinking"}
  ],
  "datePublished":"2025-09-17",
```

```
  "license":"https://creativecommons.org/licenses/by/4.0/",
  "keywords":["hybrid research","witnesses","copy-safe math","CT/HoTT"],
  "identifier":[
    {"@type":"PropertyValue","propertyID":"sha256","value":"<bundle_sha256>"},
    {"@type":"PropertyValue","propertyID":"doi","value":"<doi-or-rg-link>"}
  ],
  "isPartOf":{"@type":"CreativeWorkSeries","name":"ai1-hybrid"},
  "isBasedOn":[

{"@type":"CreativeWork","identifier":{"@type":"PropertyValue","propertyID":"sha256","value"
:"<prior_artifact_sha>"}}

  ],
  "hasPart":[

{"@type":"MediaObject","name":"paper_draft_v3.md","contentUrl":"./paper_draft_v3.md","id
entifier":{"@type":"PropertyValue","propertyID":"sha256","value":"<sha>"}},

{"@type":"MediaObject","name":"kit/manifest.json","identifier":{"@type":"PropertyValue","pr
opertyID":"sha256","value":"<sha>"}}

  ],
  "citation":[
    "Youvan, D.C. and ChatGPT. Achieving Artificial Intelligence (youvan.ai1)..."
  ],
  "subjectOf":[
    {"@type":"CreativeWork","name":"Reproducibility
Kit","identifier":{"@type":"PropertyValue","propertyID":"sha256","value":"<sha>"}}
  ]
}
```

**Series index (index/series_ai1_hybrid.jsonld).**


{

"@context":"https://schema.org",

"@type":"CreativeWorkSeries",

"name":"ai1-hybrid",

"hasPart":[

{"@type":"ScholarlyArticle","name":"Deterministic Witnesses, Generative Drafts","identifier":{"@type":"PropertyValue","propertyID":"sha256","value":"<bundle_sha>"}},

{"@type":"ScholarlyArticle","name":"Achieving Artificial Intelligence (youvan.ai1)","identifier":{"@type":"PropertyValue","propertyID":"doi","value":"10.13140/RG.2.2.36085.03047"}}

]

}


**Cross-paper linking rules.**

- **Backward edges:** isBasedOn to any artifact/paper by **hash or DOI**.
- **Forward edges:** subjectOf to follow-on kits, tutorials, or replications.
- **Sibling edges:** isPartOf the shared series; optional isSimilarTo for thematic links.

**Validation & health checks.**

- Keys sorted; UTF-8 with \n.
- No broken contentUrl inside public bundles; prefer relative paths.
- Add a jsonld_valid: true flag to run_card.json after a successful lint.

**Minimal linter (concept).**

- Confirm all identifier entries have either sha256 or doi.
- Ensure every hasPart listed appears in kit/manifest.json.
- Warn if citation points to missing or unresolvable records.

**Ship checklist (Repro Kit).**
scaffold present ✓ | seed registry ≤ 7 items ✓ | run card filled ✓ | manifest + sums consistent ✓

| counter-witnesses present ✓ | prompt provenance logged ✓ | JSON-LD valid ✓ | replay scripts run once on clean machine ✓

## 7. Metrics and KPIs

### 7.1 Reuse ratio (RR), proof density (PD)

**Unit of reuse.** A reusable unit is any deterministic artifact with a valid witness (e.g., rewrite log, test, or script) that is *imported by hash* into the current paper: code, schema, proof step, figure pipeline, or glossary block.

**Weighting.** Size/impact buckets with default weights: tiny=1, small=2, medium=3, large=5. Bucketing rule lives in kit/manifest.json as metadata.

**Formula.**

RR = (sum_i w_i * reused_i) / (sum_i w_i)

where reused_i in {0,1} for unit *i*, and w_i is its weight. Target: non-decreasing across papers in a series.

**Material claims.** A claim is “material” if it supports the abstract, conclusion, or any numbered theorem/result. Non-material = narration or orientation.

**Proof density.**

PD = proofs_or_witnesses / material_claims

Count a claim toward the numerator iff it has a linked witness: rewrite log, derivation, test, or data+code path. Target: PD >= 0.7 outside clearly labeled speculation.

**Minimal log (metrics_claims.json).**

```
{
  "material_claims": 10,
  "proofs_or_witnesses": 8,
  "rr_details": [
    {"name":"rewrite_engine_min.py","sha256":"<...>","weight":5,"reused":1},
    {"name":"copy_safe_math_guide.txt","sha256":"<...>","weight":2,"reused":1}
  ]
```

}

---

### 7.2 t_half_out and path-entropy proxy

**t_half_out (time to half-output).** Elapsed wall time from the start of **Generate** until 50% of the final text (or code lines) is stable. “Stable” = survives to publication with only whitespace or style edits.

**Computation (text).**

1. Snapshot drafts at timestamps t_k (autosave or commits).
2. For each snapshot, compute stable_tokens(t_k) = tokens that appear in the final draft with unchanged order and wording (Levenshtein distance filter ignoring whitespace).
3. Let T_final = stable_tokens(final). Then:

t_half_out = min { t_k : stable_tokens(t_k) >= 0.5 * T_final }

Target: decreases over successive papers (faster convergence).

**Computation (code).** Same idea, but use line hashes with a normalized diff (ignore comments if desired).

**Path-entropy proxy.** Measures “exploration spread” in production. Lower is better *after* early ideation.

Two-part proxy:

- **Seed entropy.** Branching over seed IDs used in Generate passes.

p_i = uses_of_seed_i / total_seed_uses

H_seed = - sum_i p_i * log2(p_i)

H_seed_norm = H_seed / log2(N_seeds_used)  // in [0,1]

- **Churn ratio.** Edit churn scaled to final length.

C = min(1, total_line_edits / final_line_count)

**Combined proxy (0..1).**

path_entropy_proxy = 0.5 * H_seed_norm + 0.5 * C

Interpretation: early papers may have higher values; aim to reduce over time without sacrificing PD.

**Minimal log (metrics_convergence.json).**


```
{
  "t_half_out_minutes": 210,
  "seed_entropy_norm": 0.42,
  "churn_ratio": 0.35,
  "path_entropy_proxy": 0.385,
  "snapshots": [
   {"t":"2025-09-17T10:00-05:00","stable_tokens":2100},
   {"t":"2025-09-17T14:00-05:00","stable_tokens":4800}
  ],
  "final":{"stable_tokens":9200,"final_line_count":1120,"total_line_edits":390}
}
```


---

**7.3 Repair/restore rates; auditability score**

**Post-publication windows.** Track at 7d, 30d, and 90d after release.

**Repair rate (errata pressure).**

repair_rate = repaired_items / published_items

- *repaired_item* = any artifact or claim that required a correction or retraction after publication (typos excluded unless they affect meaning).

**Restore rate (resilience).**

restore_rate = restored_items / attempted_replications

- *restored_item* = a previously failing replication (or flagged counter-witness) that now passes after a documented fix.

**Auditability score (0..100).** Weighted completeness of the kit.

Components (each in 0..1):

- artifact_hash_coverage = fraction of shipped files listed in SHA256SUMS.txt
- claim_witness_coverage = PD capped at 1.0

- counter_witness_coverage = fraction of material claims with a counter-witness entry
- jsonld_valid = 1 if index/entry.jsonld lints cleanly, else 0
- replay_pass = fraction of replay scripts that succeed on a clean machine

Composite:

```
auditability_score = 100 * (
  0.35*artifact_hash_coverage +
  0.25*claim_witness_coverage +
  0.15*counter_witness_coverage +
  0.15*jsonld_valid +
  0.10*replay_pass
)
```

Badges: Bronze >= 70, Silver >= 85, Gold >= 95.

**Minimal post-pub log (metrics_postpub.json).**

```
{
  "window_days": 30,
  "published_items": 42,
  "repaired_items": 2,
  "attempted_replications": 9,
  "restored_items": 7,
  "artifact_hash_coverage": 0.98,
  "claim_witness_coverage": 0.82,
  "counter_witness_coverage": 0.76,
  "jsonld_valid": 1,
  "replay_pass": 1,
  "auditability_score": 91.1
}
```

**Operational targets.**

- RR: monotone non-decreasing series; investigate if flat for 2 papers.
- PD: >= 0.7 baseline; >= 0.85 for math-heavy sections.
- t_half_out: downtrend; if it rises twice consecutively, tune seeds/process.
- path_entropy_proxy: high early, decreasing thereafter; spikes trigger a method review.
- repair_rate: < 0.1 at 30d; if higher, add pre-pub checks.
- restore_rate: >= 0.7 (fixes work); if low, prioritize root-cause analysis.
- auditability_score: >= 85 target; block publication if < 70 unless explicitly justified.

## 8. Governance and Ethics

### 8.1 Speculation labels; refusal tests; after-action review

**Speculation labels (visible in body text + kit).**
Use a 4-level, reader-facing tag at paragraph/figure granularity.

- S0: Established. Supported by witnesses/citations; no open counter-witness.
- S1: Inferred. Consistent with evidence; minor gaps; counter-witness present.
- S2: Hypothesis. Directionally plausible; requires future tests; counter-witness mandatory.
- S3: Visionary. Conceptual/aspirational; not decision-relevant; may be moved to Appendix.

**Placement rules.**

- Put S1–S3 in square brackets at the start of the claim paragraph or figure caption.
- Any section with S2–S3 must include a concrete “How this could fail” sentence and a test plan entry in counter_witness_min.json.

**Refusal tests (when to decline or downgrade).**
Refuse or downgrade content if ANY trigger fires:

- **Unverifiable harm:** biomedical dosing, legal advice, security exploits, or instructions that materially raise risk.
- **Privacy/consent gaps:** PII, medical/financial data, or non-consensual content.

- **Fabrication risk:** facts likely to have changed without current citations.
- **License conflict:** third-party assets lacking compatible terms.
  Actions: (i) refuse with a brief rationale; (ii) offer a safe alternative (general info, citations, or formalization path); (iii) log in logs/decision_log.md.

**After-Action Review (AAR).**
Run an AAR within 24–72 h of publication or any refusal override.

logs/aar.json (skeleton):


```
{
  "paper":"Deterministic Witnesses, Generative Drafts",
  "date_local":"2025-09-17",
  "timezone":"America/Chicago",
  "events":[
    {"type":"refusal","section":"5.2","reason":"license conflict","action":"replaced with illustrative sketch"},
    {"type":"speculation","section":"9.2","label":"S2","counter_witness":"C7"}
  ],
  "lessons":["Move evidence-like diagrams to deterministic pipeline.","Tighten unstable-fact checks before publish."],
  "owner":"Douglas C. Youvan"
}
```


---

### 8.2 Licensing defaults (CC BY 4.0; MIT/Apache-2.0)

**Defaults.**

- **Text (papers, captions, prompts, JSON-LD):** CC BY 4.0.
- **Code (scripts, validators, schemas):** MIT by default; Apache-2.0 if patents or contributions from multiple orgs are expected.
- **Images/diagrams:** CC BY 4.0 if wholly original; record third-party assets and their licenses in fig/LICENSE.txt.

**License hygiene rules.**

- Every repo root includes LICENSE.txt stating defaults and exceptions.
- Each non-trivial code file starts with a short license header and points to the root license.
- Third-party inclusions must be **vendored** with their license file and noted in kit/manifest.json under "third_party": [...].
- If a figure incorporates outside material (e.g., icons, textures), record source and terms; if unclear → replace or redraw.

**Dual-licensing & embargoes.**

- Dual-license code (MIT + Apache-2.0) if downstreams may prefer one; note in LICENSE.txt.
- If an external venue requires temporary embargo, clearly mark dates and scope in CITATION.cff and index/entry.jsonld and avoid releasing affected assets publicly until the embargo lifts.

**Attribution snippet (paste-ready).**



---

**8.3 Public-benefit posture and conflict disclosures**

**Public-benefit commitments.**

- **Clarity first.** Write for first-pass readability; provide plain-English summaries for each theorem/claim.
- **Witness-over-mystique.** Prefer checkable artifacts to authority claims; when not feasible, downgrade to S2–S3 and state tests.
- **Open reuse.** Favor permissive licenses; encourage forks and external replications; respond to good-faith issues via metrics_postpub.json.
- **Non-exploitation.** No paywalled critical artifacts; no dark patterns; no collection of personal data.

**Conflict disclosures (visible, concise, specific).**
Maintain a short “Disclosures” box near the end of the paper and a machine-readable copy in JSON-LD.

Disclosure checklist:

- **Funding & gifts:** grants, compute credits, data access, or sponsorships.
- **Affiliations:** current roles, board seats, or advisory relationships relevant to the topic.
- **Commercial intent:** patents filed, product roadmaps, or monetization aligned with the work.
- **Model/tool constraints:** when a specific vendor model materially shapes the result.
- **Data provenance:** source, consent status, and any restrictions.

index/disclosures.json (skeleton):

```
{
  "funding": [],
  "affiliations": [],
  "commercial_intent": "none",
  "model_constraints": ["GPT-5 Thinking used for drafting assistance"],
  "data_provenance": "No private datasets used; only publicly cited sources."
}
```

**Ethics ledger (lightweight).**
Append a one-page ledger to logs/decision_log.md when decisions affect reader trust:

2025-09-17 • §8.2 • license • Switched example code to MIT due to downstream reuse.

2025-09-17 • §5.2 • safety • Replaced data-looking diagram with reproducible plot; added hash.

**Escalation & overrides.**

- **Emergency override** (rare): If refusing would itself cause clear public harm (e.g., urgent correction of a dangerous misinformation), allow a narrow, safe disclosure with strict bounds; immediately schedule an AAR and mark the section S1 with rationale.
- **Third-party audit option:** invite external replication; if a counter-witness is sustained, issue a visible correction within 7 days.

**Publication gate (Governance & Ethics).**
Before release: speculation labels applied ✓ | refusal tests passed ✓ | licenses present and consistent ✓ | disclosures filled ✓ | AAR scheduled ✓.

## 9. Case Vignettes

### 9.1 CT/HoTT-styled symbolic AI (ai1 series)

**Goal.** Demonstrate the hybrid loop on a tiny, checkable equational core that will be reused across the ai1 series.

**Scope.** A minimal category fragment with composition, identities, and a pushout horn check. Deterministic witnesses ship with the draft.

**Deterministic claims (S0).**

- **C1. Mid-identity elimination.** g \circ id_B \circ f -> g \circ f under f: A->B, g: B->C.
- **C2. Associativity record.** (h \circ g) \circ f -> h \circ (g \circ f) where types line up.
- **C3. Pushout horn well-typedness.** If inl: A->A+B, inr: B->A+B, u: A+B->X, and u \circ inl = f, u \circ inr = g, then the mediating condition checks without free variables (type-complete log).

**Executable artifacts.**

- rules.json (named rewrites; side conditions)
- rewrite_log.json (per-step proof for C1–C3)
- validate.py (type + pattern check; NF equality)
- SHA256SUMS.txt, manifest.json, replay.sh, replay.ps1

**Witness snippet (C1, excerpt).**


```
{
  "proof_id":"C1",
  "goal":"g \\circ id_B \\circ f -> g \\circ f",
  "context":{"f":"A->B","g":"B->C","id_B":"B->B"},
  "steps":[
    {"lhs":"g \\circ id_B \\circ f","rule":"mid_identity_elim","rhs":"g \\circ f","check":"type_safe"}
```

```
 ],
 "result":"proved"
}
```

**Counter-witness (minimal).**

C1 would fail if: id_B were mis-typed or rule mid_identity_elim mismatched B.

Test: run validate.py on rewrite_log.json with rules.json; expect proved.

**Generative assist (S1).** A one-panel diagram of the pushout square labeled “Illustrative.” Caption begins with “Illustrative.” and does not assert a result.

**Operational footprint.**

- Reuse in later papers: validator, rules, harness (RR increases).
- PD remains >= 0.85 for section 4 claims (all ship with logs).

**Expected KPI movement.** Lower t_half_out on subsequent ai1 notes; path_entropy_proxy declines as seeds stabilize.

---

**9.2 Biophysics note with copy-safe derivations**

**Topic.** Fluorescence resonance energy transfer (FRET) identities packaged with deterministic checks; no clinical guidance.

**Deterministic claims (S0).**

- **C4. Lifetime identity.** E = 1 - tau_DA / tau_D, where E is dimensionless efficiency, tau_* are lifetimes with same units.
- **C5. Distance relation.** E = 1 / (1 + (r / R0)^6) with r, R0 in the same length units implies E in (0,1).

**Copy-safe math (derivation sketch).**

Given donor lifetime with acceptor tau_DA and donor-only lifetime tau_D:

E := 1 - tau_DA / tau_D.

Let R0 be the Förster radius for a given donor/acceptor pair. Then:

E := 1 / (1 + (r / R0)^6).

Dimensional checks:

[tau_DA / tau_D] = 1 (dimensionless)

[r / R0] = 1 (dimensionless)

**Executable artifacts.**

- fret_identities.json (machine-readable forms + units)
- units_validate.py (verifies dimensionless forms; fails on unit mismatch)
- synthetic_data.py (generates (r, E) pairs from chosen R0 with noise; records seed)
- plot_from_data.py (creates evidence plot; figure is *not* generative art)

**Counter-witness (minimal).**

C4 would fail if lifetimes are averaged across incompatible conditions (violates same-state assumption).

Test: units_validate.py + scenario check; mark failed if metadata differs.

C5 would fail if R0 units differ from r units or if exponent handling is altered.

Test: units_validate.py with randomized unit swaps; expect failure.

**Generative assist (S1).** A lead image evoking “energy transfer” (abstract wavelets) labeled “Illustrative.” No numeric claims.

**Out-of-scope guard.** No dosing advice, no clinical interpretation. Any biological speculation is S2 with a test plan.

**Expected KPI movement.** High PD for equations; auditability score boosted by code-generated evidence figure and unit tests.

---

### 9.3 Pedagogy artifacts (Twin Cubes, templates)

**Goal.** Classroom-ready manipulatives for CT/HoTT intuition that are easy to print, assemble, and verify.

**Artifacts.**

- fig/twin_cube_net_A4.pdf, fig/twin_cube_net_Letter.pdf (printable nets)
- templates/twin_cubes.json (faces, arrows, labels, left/right orientation)
- assembly_guide.md (short steps + photos or line drawings)
- lesson_cards.md (3–5 micro-exercises: associativity paths, identity witness, path composition)
- alt.json (accessible descriptions)
- LICENSE.txt (images CC BY 4.0)

**Deterministic checks (S0).**

- **C6. Label consistency.** Every arrow printed on the net corresponds to a well-typed edge in templates/twin_cubes.json.
- **C7. Orientation sanity.** Left-handed and right-handed nets map to mirrored face orderings; validator reports parity.

**Validator sketch.**

```
{
  "check_id":"cube_labels",
  "inputs":{"templates":"templates/twin_cubes.json","pdf":"fig/twin_cube_net_Letter.pdf"},
  "rules":[
    "all arrows in pdf appear in templates (name, src, dst)",
    "handedness parity = +/-1 matches declared orientation"
  ],
  "result":"passed"
}
```

**Speculative claim (S2, clearly labeled).**

- *"Students using Twin Cubes for 30 minutes show improved retention on path composition questions compared to a text-only control."*
  Counter-witness plan: preregister a tiny A/B classroom test; measurement rubric and anonymized results template; accept falsification.

**Generative assists (S1).**

- Cover illustration of the two cubes with subtle glow; caption begins with "Illustrative."
- Optional decorative icons on blank faces; not evidence-bearing.

**Teacher pack (repro).**

- Single ZIP with: nets (PDF), templates (JSON), validator, lesson cards, replay scripts for label checks, and SHA256SUMS.txt.

**Expected KPI movement.**

- RR rises as nets/templates carry across lessons and papers;
- PD stays high for label/orientation claims;
- Post-pub restore_rate benefits from reproducible assembly checks.

---

**Across all vignettes — publication gate.**

- Deterministic parts ship with logs, code, and hashes (PD target met).
- Generative parts are labeled "Illustrative." with provenance in prompt_provenance.jsonl.
- Each vignette has at least one counter-witness entry and a replay script that passes on a clean machine.

## 10. Limitations and Falsifiers

### 10.1 Where witnesses are thin (and what to do)

**Typical thin spots.**

- **Illustrations posing as evidence.** Concept diagrams without code/data.
- **Time-sensitive facts.** Claims likely to change (prices, policies) without current citations.

- **One-off runs.** Results lacking inputs → code → outputs with hashes.
- **Hand-wavy math.** Informal steps where typed rewrites or unit checks are feasible.
- **Quotations.** Paraphrases without sources/dates.

**Immediate downgrades (before publish).**

1. **Relabel:** Mark S1–S3 at paragraph/figure start.
2. **Trim claim:** Reduce scope to what is presently testable.
3. **Move to appendix:** If decision-irrelevant, relocate to **Speculative**.
4. **Bind a minimal witness:** Add the smallest executable check (unit/shape/dimension test, rewrite log, schema validator).
5. **Date-stamp unstable facts** or remove.

**Thin-witness ticket (per item).**


```
{
  "id":"TW-007",
  "section":"5.2",
  "claim":"Diagram suggests improved reuse",
  "label":"S2",
  "upgrade_plan":"Emit RR plot via metrics.py from reuse_log.json",
  "deadline":"2025-10-01",
  "owner":"human",
  "status":"open"
}
```


**Editorial pattern.** If a claim *could* be checked, we **must** either (i) check it now, or (ii) clearly mark it and write the test plan. No third option.

---

**10.2 When reuse fails (and how to learn)**

**Failure taxonomy.**

- **Spec drift.** Interface changed (fn(a,b) → fn(ctx,a,b)); old artifacts no longer fit.
- **Boundary mismatch.** Hidden preconditions (types, units, domains) not encoded.
- **Concealed dependency.** Unpinned package, environment knob, external asset.
- **Hidden stochasticity.** Non-seeded randomness slipped into "deterministic" path.
- **Version skew.** OS/Python/library mismatch vs. original run card.
- **License block.** Asset can't be redistributed.

**Forensics loop (24–72 h).**

1. **Reproduce original context** from run_card.json (same OS/Python); log exact diffs.
2. **Minimize** to a smallest failing case (MRE).
3. **Classify** with taxonomy above; pick a fix.
4. **Patch**: encode preconditions, pin deps, add adapters, seed RNG, or excise.
5. **Write a property/invariant test** so this failure can't recur.
6. **Record** in reuse ledger; adjust RR retrospectively if unit was wrongly counted.

**Conformance tests (add).**

- **Type/property tests:** f: A->B rejects mismatched A'.
- **Unit tests:** length/time/mass checks for formulas.
- **Idempotence/associativity** where algebraic laws are claimed.
- **Seeded determinism:** seed -> identical outputs (sha256).

**RR penalty & remediation (kit/rr_log.json).**


```
{
  "paper":"ai1-hybrid-03",
  "reused_units":12,
  "downgraded": [
```

{"name":"rewrite_engine_min.py","reason":"spec_drift","action":"adapter_v2","new_tests":["prop_assoc.json"]}

],

"rr_before":0.74,

"rr_after":0.68,

"remediation_due":"2025-10-05"

}

**When to retire a unit.**

- Two consecutive reuse failures of the same type, or
- Fix requires breaking upstream consumers without clear benefit.
  Action: mark unit retired in manifest.json; create successor with a stable adapter.

---

**10.3 External replication as the deciding test**

**Principle.** Our own replays are necessary but insufficient; *independent* replication decides credibility.

**Replication tiers.**

- **Bronze:** Third party replays with our environment + scripts; hashes match.
- **Silver:** Third party replays on a different stack (OS/toolchain) with only the kit; results equivalent.
- **Gold:** Independent team reconstructs from prose + schemas (no scripts), or extends result; counter-witness remains unsustained.

**Request-for-Replication (RFR) note (paste-ready).**

Title: Deterministic Witnesses, Generative Drafts — RFR

Scope: Verify C1–C3 (typed rewrites) and RR plot for ai1-hybrid series.

Materials: kit/ (manifest, SHA256SUMS, validate.py, metrics.py), index/entry.jsonld

Contact: <email>; Window: 30 days; Recognition: named in JSON-LD + Acknowledgments.

**Badge & JSON-LD marking (upon success).**

```
{
  "@context":"https://schema.org",
  "@type":"ScholarlyArticle",
  "name":"Deterministic Witnesses, Generative Drafts",
  "award":[
    "Replication: Bronze (2025-10-02, Team X)",
    "Replication: Silver (2025-10-12, Team Y)"
  ]
}
```

**Negative results policy.**

- Publish a **visible erratum** within 7 days if a counter-witness is sustained.
- Update counter_witness_min.json status: falsified; adjust PD; drop or rewrite affected sections.
- If core claims fall, issue a **Retraction Note** and archive the prior bundle with a **time-capsule hash** (freeze, don't edit) and a clear pointer to the corrected successor.

**Minimal replication bundle checklist.**

- kit/manifest.json, SHA256SUMS.txt (100% coverage)
- run_card.json (env + seeds)
- replay.sh / replay.ps1 (no external network calls)
- validate.py, metrics.py (property tests + RR/PD calculators)
- Small synthetic datasets with generation scripts
- index/entry.jsonld with DOI/hash identifiers

**Deciders & gates.**

- **Publish gate:** If a core claim lacks at least an internal replay, defer or downgrade to S2.
- **Series gate:** If two consecutive papers fail to achieve ≥ Bronze replication on at least one core claim, pause the series for a **methods tune-up**.

- **Close-out gate:** A claim is "established" only after ≥ Silver replication or a Gold extension.

**Culture note.** We **fail loudly and specifically**: state which claim fell, why, and what test now guards that seam. This is how the method improves.

## 11. Roadmap

### 11.1 From ai1 → ai2: broader invariants, stronger witnesses

**ai1 (now):** typed rewrites to NF; mid-identity elimination; associativity; pushout horn checks; per-claim logs + SHA256; replay scripts.

**ai2 (targeted upgrades).**

**A. Broader invariants (what the engine knows).**

- **Functoriality & naturality.** Verify F(id_A)=id_{F(A)} and F(g ∘ f)=F(g) ∘ F(f); naturality squares checked as typed rewrite goals.
- **Adjunction laws.** Unit/counit triangles as machine-checkable patterns.
- **Monoid/monad laws.** (x ⋆ e = x = e ⋆ x), associativity; return/bind coherence as logged rewrites.
- **Limits/colimits surface.** Pullback/pushout formation + universal properties expressed as existence/uniqueness rewrite obligations.
- **Higher paths (HoTT-lite).** Path concatenation/neutral element; simple homotopies as typed terms with normalization targets.

**B. Stronger witnesses (how we know).**

- **Dual proofs:** keep rewrite_log.json, and add a **mechanized check** (lean_report.json or coq_report.json) for selected theorems. If the proof assistant isn't available, the claim downgrades to S1 with a counter-witness.
- **Property-based tests.** For algebraic laws, generate typed random terms under constraints; log counterexamples (if any) with seeds.
- **Metamorphic tests.** Invariance under renaming, α-equivalence, and type-isomorphism adapters.
- **Determinism lint.** Static check that all paths are seed-fixed, filesystem-local, and network-free; produce determinism_report.json.

- **Environment pinning.** Emit env_lock.json (OS/Python/tool versions) plus requirements_lock.txt; replay refuses to run on mismatched major versions unless --allow-drift is set and logged.

**C. Acceptance gates for "ai2" label.**

- PD ≥ 0.85 on math sections, with at least 1 mechanized check per core law.
- RR non-decreasing vs. prior paper; no retired core units without successors.
- **Replication ≥ Silver** on ≥ one core claim within 30 days.
- auditability_score ≥ 90.

---

### 11.2 Community curation: seed registry → toolchains

**From seeds to toolchains.** Seeds name *intent*. Toolchains encode *how intent becomes artifacts* as a small, portable DAG.

**Toolchain spec (minimal JSON).**

```
{
  "toolchain_id":"ai2-core-laws@202509",
  "inputs":[{"name":"prep/priors.md","sha256":"<...>"}],
  "steps":[
    {"id":"gen_draft","uses":"prose:expand_outline","with":{"seed_id":"20250917-hybrid-core"}},
    {"id":"prove_assoc","uses":"proof:rewrite","with":{"goal":"(h∘g)∘f -> h∘(g∘f)"}},
    {"id":"prop_tests","uses":"test:property","with":{"law":"associativity","samples":256}},
    {"id":"mechanize","uses":"proof:lean","with":{"file":"assoc.lean"}}
  ],
  "outputs":[
    {"name":"paper_draft_vN.md"}, {"name":"rewrite_log.json"},
    {"name":"prop_test_report.json"}, {"name":"lean_report.json"}
  ]
}
```

**Curation process.**

- **Admission.** Propose a seed/toolchain via PR containing: sketch, intended scope, example outputs, and counter-witness plan.
- **Review.** Two maintainers test replay on clean machines; check license hygiene and determinism.
- **Versioning.** Semantic version for toolchains (major.minor.patch); seeds carry dates and lifecycle (draft/active/retired).
- **Deprecation.** If a toolchain increases path-entropy or lowers RR for two consecutive papers, mark **deprecated** and open a replacement ticket.

**Public registry artifacts.**

- registry/seeds.json (curated set with tags and examples)
- registry/toolchains.json (DAG specs with verified hashes)
- registry/badges.json (which toolchains achieved Bronze/Silver/Gold replication)

**Contributor kit.**

- Lint (registry_lint.py) checks schema, licenses, determinism.
- Replay harness that runs all toolchains in a sandbox and emits a unified community_replay_report.json.

**Governance minima.**

- Code of conduct; lightweight RFC process; clear review SLAs.
- “No dark prompts”: only seed sketches in public; long prompts private or redacted.

---

**11.3 Toward open, witness-centric research commons**

**Principles.**

- **Artifacts > assertions.** Every material claim anchored to code/data/derivation with hashes.
- **Commons, not silo.** Interop first: simple, text-based schemas (JSON/YAML), CC BY 4.0 (text), MIT/Apache-2.0 (code).
- **Neutral indexing.** Discoverability through JSON-LD entries that any archive can crawl.

**Core building blocks.**

1. **Content-addressed object store.** Any participant can mirror bundles; identity = sha256.
2. **Open index.** A federated list of entry.jsonld files with backlinks (isBasedOn, hasPart) and replication badges.
3. **Replay sandboxes.** Minimal containers/venvs that run replay.sh/replay.ps1 offline.
4. **Badge service (stateless).** Reads manifests/reports and issues signed Bronze/Silver/Gold replication attestations (stored alongside JSON-LD).
5. **Errata feed.** Append-only log of corrections/falsifications keyed by hash; subscribers can auto-flag local copies.

**Interoperability contract (thin).**

- **Manifest:** sorted kit/manifest.json with 100% file coverage.
- **Sums:** kit/SHA256SUMS.txt with LF newlines.
- **Run card:** kit/run_card.json capturing environment + seeds.
- **Provenance:** kit/prompt_provenance.jsonl (optional to share long prompts; sketches required).
- **Index:** index/entry.jsonld with identifier.sha256 and optional doi.

**Commons health KPIs.**

- **Median auditability_score** across hosted papers.
- **Replication velocity:** median days to first Bronze replication.
- **Reuse graph density:** average number of isBasedOn edges per paper (content-addressed).
- **Errata half-life:** time from issue discovery to visible correction.

**Participation paths.**

- **Reader → Replayer.** Download bundle; run replay; file a replication ticket (auto-generated template).
- **Replayer → Contributor.** Submit new counter-witnesses, tests, or adapters via PR; earn replication badges.

- **Contributor → Maintainer.** After N merged PRs + stable reviews, gain rights to admit seeds/toolchains.

**Sustainability & neutrality.**

- Keep core specs small and vendor-agnostic; no dependence on proprietary features for verification.
- Mirror indexes across multiple institutions or personal sites; JSON-LD makes it crawlable without central control.
- Encourage university courses to publish classroom "micro-replications" (excellent pedagogy + stress for the commons).

**End-state sketch.** A public, low-friction loop where claims arrive *already* bundled with witnesses, replication is a button-press away, and reuse is the default path—not an exception.

## Appendices

### A. Hybrid Note templates and badge spec

#### A.1 Hybrid Note (title-page block, paste-ready)

Hybrid Note — ChatGPT × youvan.ai

This article is a co-authored work by Douglas C. Youvan (human) and ChatGPT (AI).

Drafting assistance: GPT-5 Thinking; final human pass on <YYYY-MM-DD> (America/Chicago).

Deterministic components ship with witnesses (rewrite logs, code, data) and SHA256 hashes.

Generative components (prose assists, diagrams, prompts) are labeled and recorded in kit/prompt_provenance.jsonl.

Licenses: Text CC BY 4.0; Code MIT (unless otherwise noted); Images CC BY 4.0.

Reproducibility kit: see kit/ (manifest.json, SHA256SUMS.txt, replay scripts).

#### A.2 Hybrid Note (footer micro)

Hybrid: Youvan (human) × ChatGPT (GPT-5 Thinking, <YYYY-MM-DD>).

Witnesses + hashes in kit/. Generative items labeled; see prompt_provenance.jsonl.

#### A.3 Figure caption tag (for any AI-made illustration)

Illustrative (generative). Concept diagram produced with GPT-5 Thinking on <YYYY-MM-DD>.

Not evidence-bearing. See fig/lead_src.txt and kit/prompt_provenance.jsonl.

#### A.4 Machine-readable Hybrid Note (YAML stub in kit/)

```
hybrid_note:
  human: "Douglas C. Youvan"
  ai_model: "GPT-5 Thinking"
  final_human_pass_local_date: "<YYYY-MM-DD>"
  timezone: "America/Chicago"
  licenses:
    text: "CC BY 4.0"
```

```
  code: "MIT"
  images: "CC BY 4.0"
provenance_files:
  - "kit/manifest.json"
  - "kit/SHA256SUMS.txt"
  - "kit/prompt_provenance.jsonl"
```

**A.5 Badge spec**

**Purpose.** A compact, verifiable label that can be embedded in the paper or index.

**Text badge (inline, human-facing):**

[ Hybrid • ChatGPT × youvan.ai • GPT-5 Thinking • <YYYY-MM-DD> • sha256:<BUNDLE_HASH_8> ]

<BUNDLE_HASH_8> = first 8 hex chars of bundle_sha256 from kit/manifest.json.

**JSON badge (machine-facing, in kit/badge.json):**

```
{
  "badge":"hybrid",
  "model":"GPT-5 Thinking",
  "date_local":"<YYYY-MM-DD>",
  "timezone":"America/Chicago",
  "bundle_sha256":"<64-hex>",
  "short_hash":"<8-hex>",
  "series":"ai1-hybrid",
  "replication_awards":[]
}
```

**Optional SVG badge (embed or host locally):**

**JSON-LD hook (adds badge to index/entry.jsonld):**

```
{
  "award": [
    "Hybrid Badge: GPT-5 Thinking • <YYYY-MM-DD> • sha256:<8-hex>"
  ]
}
```

---

**B. Reproducibility checklist (prompt sketch, hashes, manifests)**

Use this before publishing. Keep it as kit/checklist.md (tick with x).

**B.1 Preflight**

- Aim statement and falsifiers written (prep/aims.md).
- Priors/style fixed (prep/priors.md; Copy-Safe ASCII Math).
- Sources marked for stability; web facts verified or removed.

**B.2 Deterministic artifacts**

- All math/claims that can be checked have rewrite logs or tests.
- Code runs offline; random seeds fixed; environment pinned (run_card.json).
- Figures that support claims are code-generated (not generative art).

**B.3 Generative items**

- Each diagram/lead image labeled “Illustrative (generative)”.
- Each AI assist logged in kit/prompt_provenance.jsonl (model, date, seed_id, inputs_sha256, output_sha256).
- Long prompts not embedded in paper; only seed sketches recorded.

**B.4 Hashes & manifests**

- kit/manifest.json lists every shipped file with sha256.
- kit/SHA256SUMS.txt matches on POSIX (sha256sum -c) and Windows (PowerShell Get-FileHash loop).

- Bundle hash computed over sorted manifest (bundle_sha256).

**POSIX commands:**

```
find . -type f ! -path "./.git/*" -print0 | xargs -0 sha256sum > kit/SHA256SUMS.txt

sha256sum -c kit/SHA256SUMS.txt
```

**PowerShell (timestamps):**

```
$start=Get-Date; "Start: $start"

Get-ChildItem -Recurse -File | ForEach-Object {
  $h=(Get-FileHash -Algorithm SHA256 $_.FullName).Hash.ToLower()
  "$h  $($_.FullName)" | Add-Content kit\SHA256SUMS.txt
}

Get-Content kit\SHA256SUMS.txt | ForEach-Object {
  $p=$_ -split '\s+'
  if ($p.Length -ge 2) {
    $exp=$p[0]; $file=$p[-1]
    $act=(Get-FileHash -Algorithm SHA256 $file).Hash.ToLower()
    if ($act -ne $exp) { throw "Hash mismatch: $file" }
  }
}

$end=Get-Date; "End: $end"
```

**B.5 Counter-witness & replay**

- counter_witness_min.json has entries for all material claims.
- replay.sh and replay.ps1 run clean on a fresh machine.
- determinism_report.json present (no network, fixed seeds), or item downgraded to S1 with plan.

**B.6 JSON-LD & metadata**

- index/entry.jsonld validates (identifiers include sha256 and/or doi).
- CITATION.cff present with authors, date, keywords.
- Hybrid Note inserted (title-page + footer micro); badge present.

**B.7 Governance**

- Speculation labels applied (S0–S3).
- Licenses present and consistent (text/code/images).
- Disclosures filled; AAR scheduled.

---

**C. Glossary (witness, counter-witness, content addressing, JSON-LD)**

**artifact (deterministic)** — A file or result that can be replayed from inputs via code/logs; ships with a witness and sha256.

**artifact (generative)** — AI-assisted prose or illustration that aids understanding but is not evidence; labeled and logged with provenance.

**auditability score** — Composite (0..100) measuring kit completeness: hash coverage, witness coverage, counter-witness coverage, JSON-LD validity, and replay success.

**bundle hash (bundle_sha256)** — A sha256 over the sorted kit/manifest.json entries; the canonical ID of a release bundle.

**content addressing** — Referencing artifacts by sha256:<64-hex> rather than by filename/path; enables integrity and deduplication.

**counter-witness** — A minimal falsifier entry: how a claim could fail, how to test it, current status (planned/running/sustained/falsified).

**CITATION.cff** — A lightweight citation metadata file used by repositories and indexers to format references.

**determinism report** — A check that pipelines run offline, with fixed seeds and pinned environments.

**Hybrid Note** — The authorship/provenance statement that explains the single-voice, dual-provenance method and points to the kit.

**index/entry.jsonld** — Machine-readable record of the paper and kit, using JSON-LD (schema.org) for discoverability and cross-linking.

**JSON-LD** — JSON for Linked Data; adds semantic types and relations (e.g., ScholarlyArticle, CreativeWork, hasPart, isBasedOn).

**manifest.json** — Map of logical artifact names to sha256 hashes; includes a top-level bundle_sha256.

**material claim** — A statement that underpins the abstract, conclusions, or numbered results; must be witnessed or labeled speculative.

**path-entropy proxy** — A convergence metric combining seed usage entropy and edit churn; lower is better after early ideation.

**PD (proof density)** — proofs_or_witnesses / material_claims; target >= 0.7 (or higher for math-heavy sections).

**prompt provenance** — Minimal record per AI assist: model, date, seed_id/seed_sketch, inputs_sha256, output_sha256.

**replay** — Running provided scripts to verify artifacts, hashes, and claims without network access.

**RR (reuse ratio)** — Weighted fraction of deterministic units in a paper that are reused from prior, hashed artifacts.

**run_card.json** — Environment + inputs/outputs + seeds + notes; tells a replayer "what ran, where, with what."

**seed / seed sketch** — Short, reusable intent hint (tone/constraints/structure) used to drive generation; publicly recorded; long prompts omitted.

**SHA256SUMS.txt** — A portable list of file hashes for integrity checks across OSes.

**speculation labels (S0–S3)** — Reader-facing tags: S0 Established, S1 Inferred, S2 Hypothesis, S3 Visionary.

**t_half_out** — Time from start of Generate to the point when 50% of final tokens/lines stabilize.

**toolchain** — A small DAG describing how seeds and steps produce artifacts (prose → proofs → tests → reports).

**witness** — Machine-checkable evidence for a claim: rewrite logs, unit/property tests, code+data paths, with hashes and a replay recipe.

**replication tiers** — Bronze (same stack), Silver (different stack), Gold (independent reconstruction/extension).

**References**


1. Youvan, Douglas C. *Achieving Artificial Intelligence (youvan.ai1) on a PC with Category Theory & HoTT.* ResearchGate preprint, September 2025. DOI: 10.13140/RG.2.2.36085.03047.
2. Youvan, Douglas C. *ai1-Linguo: A CT/HoTT Compliance Bench for Language on a Single PC.* ResearchGate preprint, September 2025. DOI: 10.13140/RG.2.2.31238.64324.
3. Mac Lane, Saunders. *Categories for the Working Mathematician* (2nd ed.). Springer, 1998.
4. Awodey, Steve. *Category Theory* (2nd ed.). Oxford University Press, 2010.
5. The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics.* IAS, 2013.
6. NIST. *FIPS PUB 180-4: Secure Hash Standard (SHS).* National Institute of Standards and Technology, 2015.
7. W3C. *JSON-LD 1.1—A JSON-based Serialization for Linked Data.* World Wide Web Consortium Recommendation, 2020.
8. Druskat, Stephan, et al. *CITATION File Format (CFF) 1.2.0 Specification.*, 2021.