Epago — The Open Intelligence Layer for Humans and Agents

Deep research as the engine: provable, continuous improvement of small open deep-research models

The Epago Project

Technical Whitepaper · Version 2.0 · August 2026 · github.com/EpagoFoundation/epago

Abstract. The world's information is not intelligence. Search engines *find* information. Language models *generate* answers. Neither establishes what is true — which assertion traces to which source, which result a stranger can reproduce, which system is better this month than it was last. Epago is a decentralized **Research Intelligence Network** built for that gap: a market in which many competing AI research systems discover, verify, synthesize, and continuously maintain knowledge, in which every claim is traceable to a document and every improvement has to be proven, and in which the network pays only for capability it can mechanically check. We are not building one AI researcher; we are building the market where thousands compete, and the winner has to prove it.

The largest consumers of that intelligence will not be people. They will be other AI systems: a trading agent that needs the evidence behind a position, a coding agent that needs an interface's current behavior rather than the behavior frozen into its weights, a biotech agent that needs findings published last month. An agent cannot use an answer it cannot verify, and it cannot pay for one it cannot check — so machine-readable output, source-traceable claims, and a verification trail the caller can re-derive without trusting the responder are not features to be added later. They are the requirement, and they are why this network puts verification at the base of its stack rather than above it.

Deep research is the engine of that network, not its product — and it is the right engine because deep research is a procedure, not a knowledge store. The facts an evidence-synthesis agent must report live in the documents it is given, not in its weights; what separates a usable answer from a worthless one is the discipline of decomposing a question, searching, opening sources, reconciling studies that disagree, and attributing every claim. Procedures distill into small models in a way that memorized world-knowledge does not, which is why the frontier of grounded research is already open, already strong, and already small enough to run on hardware an individual can own. It is also *static*: a laboratory ships a checkpoint and moves on, nothing makes it keep improving, and nothing establishes that a successor is genuinely better rather than tuned to the tests that would judge it.

We present Epago's foundation layer, the missing mechanism: a Bittensor subnet running a permissionless king-of-the-hill competition over grounded deep-research agents in which the exam is generated *after* challenger weights are immutably frozen, part of it drawn from per-validator secret holdouts that are published in full after rotation, and acceptance requires a 99.9% bootstrap lower confidence bound on the paired per-task difference to clear an adaptive, noise-clamped floor on each accepting validator, with coronation occurring at the first block where accepting verdicts cover a stake quorum. Every verdict is a signed, replayable function of public chain state — exactly so for the protocol (seeds, task sets, digests, statistics, coronation), and, because GPU inference is not bit-reproducible on any available stack, to within a measured noise floor for the model scores themselves, which the mechanism prices rather than assumes away: the protocol substitutes verification for trust at every step where a trusted party would otherwise appear. We describe the evaluation environment (a pinned scientific-paper corpus with local BM25 search and browse tools), the two-slice exam and its grading cascade, the statistical acceptance test and its operating characteristics, the emission schedule engineered to make genuine improvement the only profitable strategy, and a threat model mapping each attack to a structural defense.

Above that engine stands the network it makes possible: research intelligence specialized by domain, mechanical verification of citation and provenance, a living knowledge layer that keeps what is presently discarded, research that maintains itself as evidence arrives, and machine-readable research infrastructure served to AI agents. The engine runs today; the layers above it — including the agent-facing interface — are stated as architecture and roadmap, marked as such where they are described (§13), and none of them alters how validators agree. A domain is only a chain-generation contract (corpus digest, task templates, architecture pins), so expansion along domain, task, modality, and corpus-reach axes is configuration rather than code. Each duel additionally emits verified task–answer–label records over freshly published documents: reward data whose scarcity, not compute, is the binding constraint on scaling reinforcement learning. Epago claims superiority only at grounded deep research per unit of compute, never at general intelligence; §1.2 states that boundary in full, alongside the published results Epago inherits rather than owns and Epago's own current measurements.

1 Introduction

Search engines find information. Language models generate answers. Neither verifies. A search engine returns documents ranked by relevance and leaves the reader to work out what is true. A language model returns fluent prose with no accountable relationship between what it asserts and what any document says — which is why, across 20 medical questions, 69% of the references ChatGPT supplied did not exist while 95% named real authors with genuine related publications [1]. Between finding and generating sits a third thing, and it is the one knowledge work actually needs: turning information into **verified intelligence** — every claim traced to a source, every improvement proven, every result reproducible by anyone.

Epago is a decentralized Research Intelligence Network. It is a market in which many independent AI research systems compete to discover, verify, synthesize, and continuously maintain knowledge, on a substrate that pays only for capability it can mechanically check. We are not building one AI researcher. We are building the market where thousands of AI researchers compete, and where the winner has to prove it.

The largest consumers of that intelligence will not be people. They will be other AI systems: a trading agent that needs the macroeconomic evidence behind a position, a coding agent that needs an interface's current behavior rather than the behavior frozen into its training data, a biotech agent that needs findings published last month. That audience changes what an answer has to be. A person can read a paragraph and decide how much to believe it; an agent cannot. An agent cannot use an answer it cannot verify, and it certainly cannot pay for one it cannot check — so machine-readable output, source-traceable claims, and a verification trail the caller can re-derive without trusting the responder are not conveniences layered on at the end. They are the requirement, and they are why this network puts verification at the base of its stack rather than above it. The interface that serves that audience directly is roadmap rather than deployed capability (§13.6); the property it depends on is produced by the engine below it every time the engine runs.

Deep research is the engine of that network, not its product. Everything the network can eventually do — specialize by domain, verify provenance, accumulate what it has established, keep it current, serve it to other machines — rests on one prior question: can a decentralized network establish, without a trusted party, that one research system is genuinely better than another? §1.1 sets out why that question is answerable now and why nothing answers it today; §1.2 specifies the engine that does, and states exactly what it has and has not proved; §1.3 sets out the layers the engine makes possible, and marks precisely which of them exist.

1.1 Why now, and why the frontier stands still

Deep research is a procedure, not a knowledge store. The facts a research agent must report do not live in its weights; they live in the documents it is asked to read. What separates a usable answer from a worthless one is procedural: decompose the question into checkable sub-questions, search, open the sources, read them, reconcile studies that disagree, and attribute every claim to a specific document.

Parameter count mostly buys recall of memorized text. In grounded research, recall is supplied by the corpus — so parameters stop being the variable that decides quality, and the procedure starts being it.

Procedures distill into small models; memorized world-knowledge does not. This is the load-bearing asymmetry, and it is the reason "small beats big" can be true here without being true generally. A procedure is a policy over a short interface — decide what to search for, decide what to open, decide what to keep, decide when to answer — and policies of that shape compress well. An encyclopedia does not.

That asymmetry is why the capability this network needs is already open. Open-weights systems now match or exceed closed deep-research products on several agentic-research benchmarks [2], and the strongest of them are small enough to run on a single consumer GPU; §1.2 gives the table, the exact figures, and the boundary of what they do and do not show. The consequence is what matters here. The frontier of grounded research is not locked inside a laboratory: it is downloadable, it is auditable, and it fits on a machine an individual can own — which is the precondition for a permissionless market in improving it.

But that frontier is static. An open checkpoint at the frontier of deep research is a snapshot of one laboratory's training run on one day. Nothing makes it better next month, and — more damaging — nothing establishes that a successor is genuinely better rather than tuned to the tests that would be used to judge it. Progress that cannot be proved is progress that cannot be trusted, and progress that nobody is paid to make does not happen. Three failures keep it that way.

Benchmarks are contaminated. Static public test sets leak into training corpora, are memorized, and are gamed. Rebuilding a benchmark from scratch exposes it: on GSM1k, a held-out clone of GSM8k drawn to match its distribution, leading models lost up to 8 accuracy points, and the size of each model's drop tracked how reliably it could reproduce the original GSM8k text (Spearman $r^2 = 0.36$) — a memorization fingerprint rather than a reasoning deficit [3]. Benchmark authors have answered with freshness and with secrecy: LiveBench and LiveCodeBench continuously retire questions in favor of tasks that postdate model training [4], [5], and MMLU-CF withholds its test split entirely, on the finding that contamination otherwise inflates scores on any fully open set [6]. Neither removes the incentive to game, only its easiest expression — even human-preference leaderboards fall to selective disclosure, with one provider privately testing 27 model variants before a major release and publishing only the best [7]. A rising score on a public suite no longer certifies a rising capability, and the field's central quality signal is broken precisely where it matters most — at the frontier of claimed performance, where the incentive to game is highest.

Evaluation requires a trusted operator. Every leaderboard and competition has someone you must believe: a company API, a private test server, a judging panel. If that party is careless, biased, or compromised, every downstream result is worthless, and outsiders cannot check.

Model development is centralized. Frontier-relevant training increasingly concentrates in a few laboratories; forecasts put the top two to three labs at 15–20% of global AI compute by 2027 [8]. An outsider with a better training idea has no venue where proving it is both permitted and rewarded.

These are one problem wearing three costumes: evaluation that cannot be verified. Remove that, and continuous open improvement becomes mechanically possible; leave it, and the open frontier stays a series of disconnected snapshots.

1.2 The engine, and what it proves

Epago's foundation layer is the mechanism that makes the small open frontier move. It is an open, permissionless competition in which a challenger takes the crown only by proving an improvement on an exam that cannot be gamed: generated *after* the competing models are frozen, half of it drawn from a private holdout of documents published after training, graded mechanically against pre-proven keys, and replayable by any stranger from public chain state. The competition is permissionless, the exam is unpredictable at submission time, and every verdict is checkable without trusting anyone — its derivation exactly, its scores against a measured noise floor (§8). This paper specifies that protocol

as implemented; §3 through §12 are its complete specification, and it is the layer every layer above inherits from.

The starting point is measured, not conjectured. Tongyi-DeepResearch-30B-A3B is an open-weights mixture of experts with 30.5B total parameters that activates only ~3.3B per token. On the agentic deep-research benchmarks its authors report, it exceeds OpenAI o3, DeepSeek-V3.1 (671B), and Claude-4-Sonnet on five of seven, and o3 still leads one of them (Table 1) [9]. Two readings of that table would be wrong. It does not say a 30.5B mixture of experts is more intelligent than a 671B one — it says that on the narrow axis of *executing a research procedure over documents the model is given*, the parameter advantage does not convert. And it is not Epago's result: it is the published result of the base model Epago begins from — the floor, not the claim.

Agentic deep-research benchmark	**Tongyi-DR-30B-A3B** (~3.3B active)	**o3**	**DeepSeek-V3.1** (671B)
Humanity's Last Exam [10]	**32.9**	24.9	29.8
FRAMES [11]	**90.6**	84.0	83.7
xbench-DeepSearch [12]	**75.0**	67.0	71.0
BrowseComp [13]	43.4	**49.7**	–

Table 1: The floor Epago starts from — published results for the genesis *base model*, not Epago's own [9]. Bold marks the row leader; a dash marks a comparator figure not restated here. The same source reports GAIA 70.9, WebWalkerQA 72.2, and BrowseComp-ZH 46.7 for the base model; across all seven benchmarks it leads five, and proprietary o3 leads BrowseComp, 49.7 to 43.4. Claude-4-Sonnet scores 20.3 / 80.7 / 65.0 on the first three rows. No Epago-trained descendant may claim any of these numbers until it has been crowned by the procedure of §7.

Because the base model activates ~3.3B parameters per token and runs at 4-bit quantization on a single 24 GB GPU in Epago's own deployment measurements, one consumer-class machine suffices to fine-tune a challenger or to run a validator — and replaying a verdict's derivation needs no GPU at all (§8). Smallness is not an aesthetic preference here; it is the condition under which decentralization is economically real rather than nominal.

A paper that condemns selective reporting has to state its own boundaries alongside its results.

> **What is claimed.** Superiority at *grounded deep research, per unit of compute* — executing a research procedure over documents supplied at inference time. **What is not claimed.** General intelligence superiority over frontier models. Outside the grounded-research regime, larger models remain ahead, and nothing in this paper contradicts that.
>
> **Whose results these are.** Every figure in Table 1 belongs to the published evaluation of the *base model*, Tongyi-DeepResearch-30B-A3B [9]. They describe the floor the competition starts from. An Epago-trained descendant inherits none of them until it is crowned (§7), at which point what it has proved is a paired improvement over its predecessor — not a benchmark score.
>
> **The loss is reported with the wins.** On BrowseComp, o3 leads the base model 49.7 to 43.4 [9]. Reporting only the favorable rows is exactly the behavior this protocol exists to make unprofitable.
>
> **"Small" means sparse, not tiny.** ~3.3B *active* parameters per token out of 30.5B total in a mixture of experts. Memory footprint is that of a 30.5B model (hence the quantization); per-token compute is that of a ~3.3B one. This is never a 3B dense model.

The measured starting point is not flattering, and that is the point. On Epago's own exam over the pinned corpus, the genesis king currently scores **~22%** with full agentic tooling. The two bounds around that number are what make it meaningful. Hand the model the evidence it would otherwise have to find and it answers **91.0%**, so the answer keys are right, the questions are answerable, and every unsolved task is measured headroom rather than a broken item. Remove the corpus entirely and

it scores **0.0%**: nothing in this exam is answerable from memory, so there is no contamination to inherit and no points available without doing the research (§5.4). Fully solvable with the evidence, wholly unsolvable without it. Those numbers are the honest description of the starting line, and they are not comparable to public leaderboard scores — the exam is deliberately constructed so that answers cannot be found, only derived. The gap from 22% to the 91% ceiling is the headroom the competition exists to close, and the per-episode telemetry names the cheapest first targets on the board: episodes that ran out of turn or context budget mid-research, and elimination searches abandoned before the last constraint was checked. An earlier draft of this exam scored the same model at 82% and was discarded for exactly that reason — §5.4 documents the measured failure and the three validity tests any future release must pass.

1.3 The network the engine makes possible

The engine selects models. What it *emits* every time it runs is something else: verified, dated, cited research artifacts over documents newer than the models being tested, each one graded against a key proven to exist in a pinned corpus, each one replayable by a stranger (§8, §14). A leaderboard throws that away. Stacked instead of discarded, it is the substrate of a network — and the network has a shape, six layers deep, in which each layer consumes what the layer below it proves.

Layer	What it is	Status
L0	**The provable-improvement engine.** Open models compete in paired duels on an exam generated after weights freeze, half of it from a private holdout of documents published after training, graded mechanically, replayable by anyone.	Runs today; specified in §3–§12
L1	**Specialized research intelligence.** A domain is a chain generation with its own corpus, its own king, and its own market, admitted on whether its answers are mechanically checkable and its documents arrive on a schedule.	Genesis generation runs today; further domains are contract, not code (§13.2)
L2	**Verification.** Citation and provenance checking as a task family: does the cited source actually contain the value claimed of it? Mechanically checkable, therefore admissible.	Design stage (§13.3)
L3	**The living knowledge layer.** Verified results accumulate as claim → evidence → source → confidence → timestamp → relationships instead of being discarded after each answer.	Planned (§13.4)
L4	**Continuous research.** Standing research objectives that update themselves as new evidence publishes, rather than one-shot reports.	Planned (§13.5)
L5	**Machine-to-machine intelligence.** Verified, cited, machine-readable research served to other AI agents through an API — intelligence infrastructure, not a consumer app.	Planned (§13.6)

Table 2: The layered architecture, each layer with its honest status. §13 states the stack in full and is, with the roadmap of §16, where this paper marks its own registers explicitly.

The engine runs today; everything above it is architecture and roadmap, described as such in §13. One rule binds the stack together: **no layer above L0 is permitted to change how validators agree.** The acceptance test of §6, the quorum rule of §7, and the audit architecture of §8 are fixed points that the upper layers are designed around, not adjustments the upper layers are allowed to make. Where a layer would require such a change, §13 says so explicitly and the layer stays on the horizon.

What the network is paid to do. The task class the competition scores is stated in §2: grounded deep research over a pinned corpus — identify the one source satisfying constraints that each match a crowd, extract exact values under stated rules, compare findings across named documents, and compute answers that exist verbatim in no document at all, with the question never identifying its sources. It is defined there without reference to any particular field, because the selection machinery never sees the field: a domain is a chain-generation contract rather than a code change (§13.2).

1.4 Why Bittensor

The alternative to trusting one centralized research AI is not a better-audited centralized research AI. It is many independent research systems competing under a rule that pays for what can be verified: who is more accurate on tasks nobody could have trained on, who grounds answers in sources that actually exist, who hallucinates less, who investigates deeper. Emissions buy measured capability, not claims.

Bittensor supplies exactly the primitives that market needs and nothing Epago would otherwise have to build for itself: an open registry of independent participants with capital at stake, an unforgeable ordering and timing service (which submission revealed first; no verdict backdated), and a native, continuous emission channel allocated against a weight vector every participant can compute for themselves [14]. What the subnet supplies in return is the thing the substrate cannot supply — a definition of what it will pay for. Epago's answer is narrow on purpose: it pays for a paired improvement that clears a statistical floor on a freshly generated exam, and for nothing else.

That narrowness is what makes the arrangement work permissionlessly. Because the exam is mechanical and the verdict is a pure function of public chain state, the network never has to ask its participants what is true, or which of them is best — only to run duels that any other participant can re-run (§7, §8). A market that had to poll its members for the truth would be a market that could be captured by whoever held the most of it. This one cannot: the competitors supply models, and the arithmetic supplies the verdict.

1.5 Contributions

1. A **paired-duel selection primitive** (§4) that compares two checkpoints on identical, freshly generated tasks, cancelling exam difficulty exactly and reducing model selection to a single scalar decision.
2. A **two-slice exam** (§5) combining commit-reveal–seeded public tasks (unpredictable before weight freeze, replayable forever after) with per-validator private holdouts under *delayed transparency* (secret while in use, published in full at rotation).
3. A **statistical acceptance test** (§6): a one-sided 99.9% bootstrap lower confidence bound over the paired difference vector, compared against a floor that adapts to the king's remaining headroom and is clamped above the measured cross-hardware noise level.
4. **Stake-quorum coronation** (§7): the champion is a pure function of public chain state, computable identically by every observer, with no coronation authority — and, as a corollary, validator weight-copying is neutralized as an attack on model selection.
5. A **fully replayable audit architecture** (§8) and an **emission schedule** (§9) whose decay, bonus, and cooldown terms are chosen so that revealing a genuine improvement immediately is the dominant strategy.
6. A **domain-agnostic generation contract** (§13.2) that makes a new domain, a harder rung of the task ladder, a new modality, or a wider corpus a configuration change rather than a protocol change.
7. **Verified reward data as a first-class output** (§14): every duel emits mechanically labelled trajectories over documents newer than the models being tested, published on a fixed schedule against pre-committed digests.
8. A **layered architecture for a research intelligence network** (§13, Table 2), stated with each layer's honest status — L0 running today, L1 configuration rather than code, L2 at design stage and deliberately scoped to what is mechanically checkable, L3 through L5 roadmap — and a fixed-point rule that forbids any upper layer from altering how validators agree.

2 The task: grounded evidence extraction over a pinned corpus

2.1 The procedure

A grounded deep-research agent is not a chatbot answering from parametric memory. Given a hard question about a body of published literature, it plans and executes a multi-step investigation over a pinned document corpus using two tools: **Search**(query) → ranked documents with snippets, and **Browse**(doc_id) → full document text. The agent interleaves reasoning and tool calls until it can commit to a short, checkable answer whose every element traces back to a specific document.

The task class is defined by what can be checked, not by subject matter. Stated in full, it is: identify the right source among many similar candidates when the question never names it, extract exact values under stated rules, compare findings across documents that need not agree, and compute answers — counts, rankings — that appear verbatim in no document, so they can only be derived from the evidence. That shape exists wherever claims must trace to sources and the answers can be mechanically checked against those sources — the scientific literature, case law, patents, regulatory filings, financial disclosure. No mechanism in §4 through §11 depends on which of them a generation is instantiated over, because the only object the selection machinery ever handles is a vector of per-task correctness differences; the corpus, the task templates, and the tool interface are contract parameters (§13.2).

The capability decomposes into retrieval, reading, extraction and combination, and the exam is constructed so that a task cannot be answered without all four. The current family is the **intersection** form: two anchor studies each name a hidden term, and exactly one further study in the corpus involves both. Difficulty is a structural ladder rather than a stylistic one — the anchors may be given by title or only by description, producing one-, two- and three-hop tiers minted at a 35/35/30 mixture — and it is measured rather than asserted: every arm tested degrades monotonically as the hop count rises (§5.4). What never varies across tiers is the property that makes the family sound, namely that the target is described nowhere in the question.

To make the demand concrete, here is one instance of the class, drawn verbatim from an audited pool. *The study titled "Microplastic contamination in thirty commercially important fish species: Distribution, polymer composition, pollution indices, and human health risks." names a specific pollution index. The study titled "Self-Powered SiC-Based Photoelectrochemical Ultraviolet Photodetectors for Robust Underwater Optical Communication Against Full Aquatic Environments" names a specific photodetector type. Exactly one other study in this corpus involves both the pollution index named in the first and the photodetector type named in the second. Give the exact title of that study.* The decisive property is that **the answer is described nowhere in the question**: no phrase in it is a query for the target. A solver must open the first paper, read the index's name out of it, open the second, read the photodetector type out of it, and only then search for the two together — three hops, each a real act of retrieval, reading and extraction. The promise that exactly one study qualifies is not rhetoric but a proved fact about the corpus, established before the sentence was written (§5.1). Read structurally, the same shape recurs across domains: an answer identified only by the intersection of properties held by two other dated, citable documents is what a patent novelty search, a case-law survey, and a regulatory-precedent search all are.

The task class compresses real evidence labor; it is not a synthetic puzzle. That is easiest to show against a professional workflow whose steps someone has already written down. A systematic review states its question in PICO terms — population, intervention, comparator, outcome — screens thousands of titles and abstracts against those eligibility criteria, extracts a fixed set of data items from the studies that survive (design, sample size, follow-up duration, effect estimates and their confidence intervals), appraises each study's risk of bias, and only then pools the evidence; PRISMA 2020 requires every one of those steps to be reported in enough detail that a reader can reproduce it [15]. The five skills above are those stages stated as answerable questions: eligibility screening is aggregation, study identification is direct retrieval, evidence-table construction is enumeration and statistics, and corroborating a reported effect against independent reports is cross-verification — the example above is a screening-and-identification step compressed into a single query. That workflow is cited as a well-documented representative of rigorous evidence work rather than as a destination: every field that has such a standard has the same skeleton — a stated question, an eligibility rule, an extraction schedule, an appraisal step, and a reporting requirement that the result be reproducible. Extraction is deliberately the *first* rung, chosen because its answer keys are mechanically checkable; §13.2 sets out the ladder from here to screening, synthesis, appraisal, and drafted sections.

2.2 Why this task class is the right thing to pay for

The competition pays for grounded evidence extraction and for nothing else. Three properties of the task class explain that choice, and none of them belongs to any one field.

Ungrounded generation fabricates its evidence base. A model answering from parametric memory cannot be relied on for provenance, and the failure is measured rather than anecdotal. In one audit of 20 medical questions, 69% of the references ChatGPT supplied did not exist, yet 95% named real authors with genuine related publications — fabrications built to survive a glance [1]; independent audits of ChatGPT-generated bibliographies report the same failure mode [16]. An audit of GPT-4o across six literature reviews found 19.9% of its citations wholly invented and 45.4% of the genuine ones erroneous, with 64% of the fabricated DOIs resolving to real but unrelated papers [17]. In one survey of clinicians, 91.8% reported encountering hallucinations in model output and 84.7% judged them capable of causing harm; domain fine-tuning alone does not remove them [18]. Grounding measurably does: constraining a model to retrieved sources lifted microbiology-QA accuracy from 71.4% to 90% for GPT-4 and from 58.6% to 92.9% for Claude 3 Sonnet [19]. Epago's task is therefore defined so that every element of an answer must trace to a specific document in the pinned corpus and is scored against it — an answer the agent cannot ground is wrong by construction, and no fluent guess is ever rewarded.

The labor it compresses is expensive, slow, and — unusually — verifiable. Where rigorous evidence work has been costed, the numbers are large: roughly 29,000 systematic reviews are published per year [20], each consuming a mean of about $141,000 in labor — 1.72 scientist-years [21] — and a mean of 67.3 weeks from registration to publication [22]. They are cited as evidence of what disciplined evidence work costs in a field that happens to measure its own costs, not as a market this paper claims. What matters here is the other half of the description: the work is *verifiable*. Its outputs are structured, its sources are citable, and its correctness is checkable against the documents rather than against a reviewer's taste — which is the property that makes a task admissible to this protocol at all (§13.2), and the property that most professional judgment work lacks.

Its most demanding users cannot hand their documents to a hosted endpoint. Evidence work of this kind is frequently done inside a regulatory perimeter, and the perimeters are not soft ones. A vendor handling protected health information becomes a HIPAA business associate, with penalties reaching $2,190,294 per violation category per year [23]; sensitive categories such as health data fall under GDPR Article 9, carrying penalties to €20M or 4% of global turnover [24], and the EDPB declines to presume that a model trained on personal data is thereby anonymous [25]; and decision-support systems in several professional settings are high-risk under the EU AI Act, which attaches logging, traceability, and human-oversight duties [26]. A reproducibility obligation runs alongside the confidentiality one: reporting standards increasingly make the system that produced a result part of the reportable method — PRISMA 2020 demands that every step of a review be reported in reproducible detail [15], and the 2025 RAISE guidance for AI-assisted evidence synthesis makes the model version itself reportable [27]. An open-weights model pinned by content digest, small enough to run inside the user's own perimeter, and selected by a procedure any stranger can replay from public data (§8) satisfies both obligations by construction; a silently updated hosted endpoint satisfies neither. Wherever a result must be reproducible down to the system that produced it, an open, self-hostable, replayable system is the only kind that qualifies — and that is a structural opening a small open model has and a large closed one does not.

These three properties travel together, and they are not confined to one field. Wherever a body of dated, citable documents supports expensive expert labor whose conclusions can be checked against the documents themselves, the same task class exists and the same mechanism applies unmodified. That is exactly the admission criterion §13.2 states for a chain generation, and exactly why the engine never needs to know what it is reading.

3 The evaluation environment

Duels run inside a pinned, local, deterministic environment. The genesis king — Alibaba's Tongyi-DeepResearch-30B-A3B (Apache-2.0), a Qwen3-30B-A3B-derived mixture of experts with 30.5B total parameters, ~3.3B activated per token, and a 128K-token context window [9] — competes over a pinned scientific-paper corpus that Epago re-hosts as its genesis-generation contract:

- **Corpus.** A pinned slice of the scientific literature, assembled from open full-text and abstract sources. The snapshot is content-addressed: its `sha256` digest is pinned in the generation contract, and a validator whose local bytes do not reproduce the digest refuses to evaluate (§8). It is a fixed, auditable set of papers — not a mirror of the open web. Which slice is pinned is a contract parameter, not a property of the engine (§13.2).
- **Search.** Local lexical retrieval via BM25 over an SQLite FTS5 index, byte-identical across validators, returning ranked papers with snippets in milliseconds — at zero marginal cost and with no live API in the loop.
- **Browse.** Full stored paper text served locally in milliseconds.

Two properties are load-bearing. First, **determinism and cost**: because the corpus is local and fixed and search is a deterministic BM25 pass, a duel of hundreds of multi-turn rollouts incurs no API fees and runs against a byte-identical world on every validator — the precondition for reproducible verdicts. Second, **leak-proof construction**: a task's question is checked against the live search backend at mint time and discarded if any string in it surfaces its source documents — measured across the current release, the question pins its source for 0% of tasks and the answer appears on the first results page for 0%, against 98.9% and 74.1% respectively for the naive cloze-style construction this release replaced (§5.4). Where a template additionally requires it, per-task masking hides linking documents from search during that task's rollouts; but the primary defense is that the question itself gives nothing away.

Task synthesis is governed by one rule, adopted after the failure mode it prevents was measured rather than imagined (§5.4): **hard to find, easy to check.** The question must never contain a string that identifies its source documents, and the answer must never be copyable out of a search result — it is reached by real research and then checked mechanically against a key the generator re-derives from the corpus. Three task families implement the rule. *Constrained search*: the question states three to five constraints — a value range, a stated sample size, words the paper must and must not contain — each of which matches a crowd of at least five papers, while only their conjunction (proven unique by a full scan at mint time) matches one; the answer is that paper's title, or a value from it that no constraint mentions. *Cross-study comparison*: the question names four to six papers by title and asks which reports the highest or lowest value of a stated quantity under a stated extraction rule; the winner is written nowhere. *Computed evidence*: the same shape, but the answer is a count over the named papers — a number that appears verbatim in no document at all. Every mint self-checks before a task is admitted: the assembled question is pasted into the actual search backend and the task is discarded if its source surfaces near the top; computed answers must round-trip through an independent recount; ranges and thresholds are placed clear of every extracted value so grading can never turn on a rounding accident.

4 Protocol overview

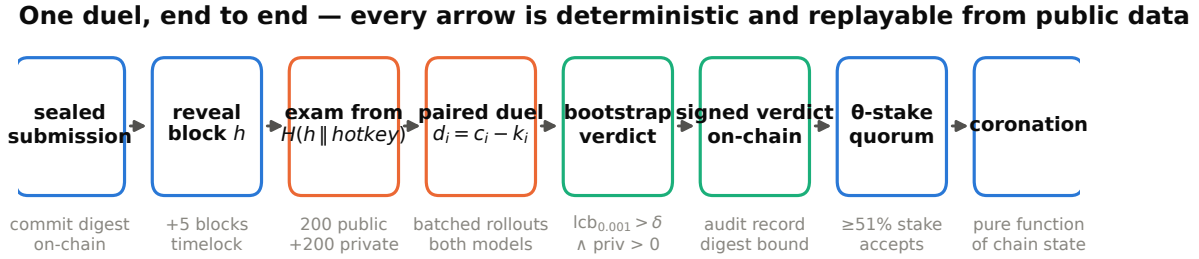

Figure 1: The duel lifecycle. Every arrow except the rollouts themselves is a deterministic function of public chain state and pinned configuration; every artifact is committed by digest. A stranger can replay the pipeline from public data — exactly, up to and after the scoring step, and statistically across it (§8).

At any time the network has exactly one **king** — the reigning champion checkpoint. Anyone may fine-tune the king (or train from scratch; the protocol never inspects provenance) and submit the result as a **challenger**. Independent **validators** each run the identical open-source duel engine; a challenger

that provably beats the king on a stake quorum of validators becomes the new king and begins earning (Figure 1).

Why paired duels. A duel scores only the per-task difference

$$d_i = c_i - k_i \in \{-1, 0, +1\},$$

where $c_i, k_i \in \{0, 1\}$ record whether challenger and king answered task i correctly. Absolute scores depend on exam difficulty, and every duel draws a fresh exam; but the difference on the *same* tasks cancels difficulty exactly — a hard exam depresses both models together and leaves d_i untouched. Rating ladders and multi-metric panels were considered and rejected: they inject cross-exam rank noise, invite metric gaming, and answer a question ("how good is everyone?") the network does not need answered. One scalar decides succession.

Identity. The author of a submission is its revealing *hotkey*; exclusivity, reign accounting, and repository naming are all keyed to it, with Sybil cost anchored by each fresh hotkey's own registration burn (§9). A hotkey carries **exactly one submission, permanently**: trying again means registering another hotkey and paying its burn, which prices every attempt rather than leaving the queue free to flood.

Submission privacy. A challenger may be submitted publicly through an open repository, or privately into the validator's own object store so that a checkpoint which has not yet won stays unreadable by its rivals. Private submission needs the miner to hold write credentials and there is no private channel to deliver them over — any such channel would reintroduce an operator, which the design forbids. So the channel is public and the payload is not: the validator publishes one mailbox file containing a credential envelope per registered miner, each sealed to that miner's hotkey and signed before sealing, each carrying a prefix-scoped, write-only, expiring credential. Everyone can read the mailbox; each miner can open exactly one entry. Crowned models are published in the open regardless, so the model the network pays is always auditable by anyone.

5 Exam construction: the two-slice exam

Each duel poses the same $N \approx 1,000$ tasks to both models (default 800 public + 200 private). Each slice defeats a different attack.

5.1 Slice 1 — public tasks: unpredictable, then permanently checkable

A challenger submission is committed on-chain under a timelock and unsealed five blocks later at its **reveal block** h. Which public tasks are asked derives deterministically from that block's hash:

$$\text{seed} = \text{blake2b}(H_h \parallel \text{hotkey} \parallel \text{public}) \rightarrow \text{PCG64} \rightarrow \text{task selection}.$$

The hash H_h does not exist until after the challenger's weights are immutably frozen (their digest is part of the sealed commitment), so **no party — submitter, validator, or protocol author — can train on the exam**. This is the commit-reveal analogue of the freshness discipline used by live benchmarks [4], [5], with a chain in place of a calendar.

How the exam is then re-derived depends on the release. A **template release** generates its questions from the seed and the pinned corpus, so anyone regenerates the identical set outright. A **sealed-pool release** — whose questions a language model worded, and which therefore cannot be a pure function of a seed, since no promise about temperature survives a provider changing hardware or model version — obtains the same two guarantees by sequencing, through three artifacts:

- the **pool**, every task with its answer, committed by digest before any round opens and kept sealed while it is in service;
- the **manifest**, the pool's task identifiers and nothing else, committed by digest and published immediately, because an opaque identifier list is useless for training yet sufficient to reproduce a selection;
- the **round file**, the tasks one round actually asked, published in full after the disclosure delay so any verdict can be re-graded.

Both digests are fixed in the generation contract **before** a round opens, so the exam provably existed before the challenger's weights were frozen and neither artifact can be substituted afterwards. Because selection runs over the sorted identifier list alone, the manifest is enough for a stranger to reproduce exactly which tasks a round drew while every unasked task stays sealed for later rounds — full verifiability and a long-lived pool at once. Publishing the whole pool after each round would be the obvious alternative and is the wrong one: a miner would then hold every question and answer, and a pool would survive exactly one round.

Rounds are disjoint. A round's tasks are retired from the pool the moment they are staged for release, so a published task is never asked again and the public record can grow without ever weakening a future exam. Retirement is by content-addressed identifier, so it holds across pool rotations: a task re-minted into a later pool keeps the identifier it was published under and stays retired.

5.2 Slice 2 — private tasks: the overfitting tripwire

The generator is open source, so a miner could overfit its *distribution* rather than improve. Each validator therefore maintains a **private pool**: tasks minted from its own secret sampling of a large, dated feed of newly indexed papers, using a locally generated secret seed. Acceptance requires the private-half mean to favor the challenger *on every accepting validator*; to overfit the network, a miner must simultaneously overfit every validator's independent secret holdout.

Private evaluation ordinarily destroys auditability — "trust my secret test set" is exactly the trust Epago refuses. The resolution is **delayed transparency** (Figure 2): the pool's digest is committed on-chain with every verdict that used it, and the full pool is published when it rotates (approximately every six days; 43,200 blocks). Every private verdict thereby becomes retroactively and cryptographically checkable; a validator that fabricated private results is caught by anyone, permanently. The pool feed itself is fully automated — a bounded, revision-pinned slice of newly indexed papers selected by secret seed — so freshness does not depend on any human curator, and evidence that postdates a challenger's training cannot have been memorized by it.

Contamination in professional corpora is measured, not hypothetical. One contamination-free benchmark built from clinical material postdating model training cutoffs found that 84% of the models it evaluated degrade on the fresh cases, the best scoring only 39.2% — consistent with pervasive leakage in the static exams in common use, including those derived from licensing question banks [28]. A model must therefore be re-proved on evidence it cannot have seen, which is exactly what the private-pool feed supplies — and it is why a steady arrival of dated documents is an admission criterion for every domain, not a convenience (§13.2).

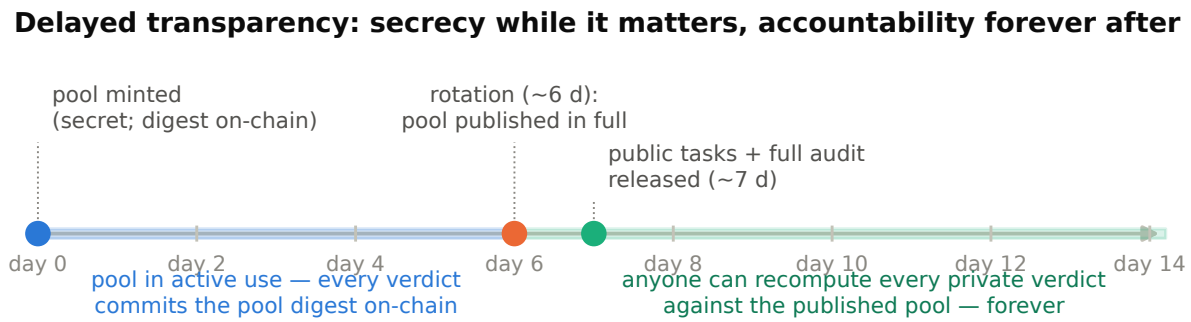

Figure 2: Delayed transparency. A private pool is secret exactly while it can influence verdicts; at rotation it is published in full against its on-chain digest, making every past private verdict independently recomputable.

5.3 Rollout and grading

Each model answers each task in an agentic rollout — up to 40 turns, 32K-token context, a 15-minute wall clock, greedy decoding at a pinned seed with a mild pinned repetition penalty, answers capped at 200 characters — speaking the tool-calling convention agentic checkpoints are actually trained on: tools declared as JSON signatures, called via structured tool-call turns, results returned as a ranked, web-

style results page (title, url, snippet) with documents opened by url. The protocol choice is load-bearing and was measured, not assumed: under a bespoke command syntax the reference model lost 48% of episodes to malformed actions and scored below its own closed-book floor; under its native convention the same model on the same tasks recovered +24 points (§5.4). Any task's masked_doc_ids are hidden from search during its rollouts where the template requires it. Grading is a cascade that is programmatic first: normalized exact match → alias match → numeric tolerance → an *optional, disabled-by-default* pinned and injection-hardened LLM judge as last resort. The judge invocation rate is published per duel; near zero certifies the exam is cleanly gradeable, and any rise is a public alarm. Because each answer key is verified to exist in the corpus at generation time, "correct" is a string comparison against a pre-proven key — never an opinion.

5.4 The exam is validated, not assumed

An exam can be private, fresh, and mechanically graded and still measure nothing. Epago treats exam validity as an empirical property to be measured, and the current family is the product of that discipline. Two instrumented numbers establish it, and three standing tests keep it established.

The task is answerable. Supplied with both anchor papers — the evidence a solver would otherwise have to find — a model answers **91.0%** correctly. This single measurement validates the pool: the answer keys are right, the questions are answerable, and the intended route genuinely reaches the intended answer. It is what converts an unsolved task from a suspected defect into measured headroom.

The task is not answerable from memory. With the tools removed entirely, the same model scores **0.0%**. There is no contamination, no pretraining leakage, and no path to a point that does not run through searching, reading and combining. Taken together the pair makes the strongest statement an evaluation can make: fully solvable with the evidence, wholly unsolvable without it.

Task soundness is independently re-derivable. Every property that makes a task correct is arithmetic over the pinned corpus, so a pool is auditable without a model, an API key or a GPU. `verify_pool` re-extracts every task's two terms across all 50,420 documents and re-derives twelve claims per task — uniqueness, concealment, label support, route takeability, content-addressed identity — comparing each against the minter's own assertion rather than trusting it. On the audited genesis pool the result is **400 of 400 verified**, audit id `sha256:45e1a468…`, with no minter-supplied file trusted anywhere in the check.

Three tests are standing infrastructure, re-run whenever the exam changes:

- **The shortcut ablation.** The same task set is scored under normal tools, oracle retrieval, pasted evidence, and no corpus at all. A valid exam produces a ladder, and this family produces an unusually wide one: 0.0% closed-book against 91.0% under oracle retrieval, so finding the evidence is worth essentially the entire score.
- **Transfer.** A change that helps the exam must also help an external benchmark, and vice versa. The native-protocol harness moved the exam +24 points and FRAMES +2.4 points together, paired on identical items in both cases.
- **The ranking test.** An honestly different model must be ordered the same way by the exam and by external benchmarks.

The difficulty ladder is likewise measured rather than declared. Across every arm tested, accuracy falls monotonically as the structural hop count rises — 24.3%, 23.6% and 18.3% for the reference model across the one-, two- and three-hop tiers — so the tiers are graded difficulty rather than labels.

One further tripwire runs in production: at every coronation the new king is automatically evaluated on a pinned external benchmark with the benchmark's own corpus behind the tools, and the result — keyed to the king's digest — is published in the audit trail beside the crown. A reign whose exam scores rise while its external anchor stays flat is publicly visible within hours, and the anchor is observational by construction: it can never grant a crown, so inflating it buys nothing.

6 The statistical verdict

A duel produces a difference vector $d = (d_1, ..., d_N)$. The question is not whether its mean is positive, but whether the challenger is *reliably* better — or merely drew a friendly exam.

6.1 Bootstrap lower confidence bound

Epago resamples the collected vector d with replacement $B = 10,000$ times (pure arithmetic; no model is re-run), computes each resample's mean, and takes the 0.1st percentile — a one-sided **99.9% lower confidence bound**, $\text{lcb}_{0.001}$. Intuitively,

$$\text{lcb} \approx \hat{\mu} - (\text{penalty for the uncertainty of } \hat{\mu}),$$

so consistent wins on a large sample keep the bound near the mean, while scattered results collapse it toward zero. Figure 3 shows two challengers with the *same true edge* receiving opposite verdicts: the bound prices inconsistency, not just magnitude. A false coronation requires a one-in-a-thousand statistical fluke per validator, compounded across the quorum — and then it must happen *twice*: the round's provisional winner is re-dueled once on a fresh confirmation exam, minted from a separate seed domain of the same block hash, and must clear the floor again before any validator commits an acceptance. Squaring the tail per attempt is what keeps the bar meaningful against a fleet of lottery tickets, without raising what a genuinely better model must beat; an unconfirmed winner settles as a near-miss, keeping its arena credit and its retry.

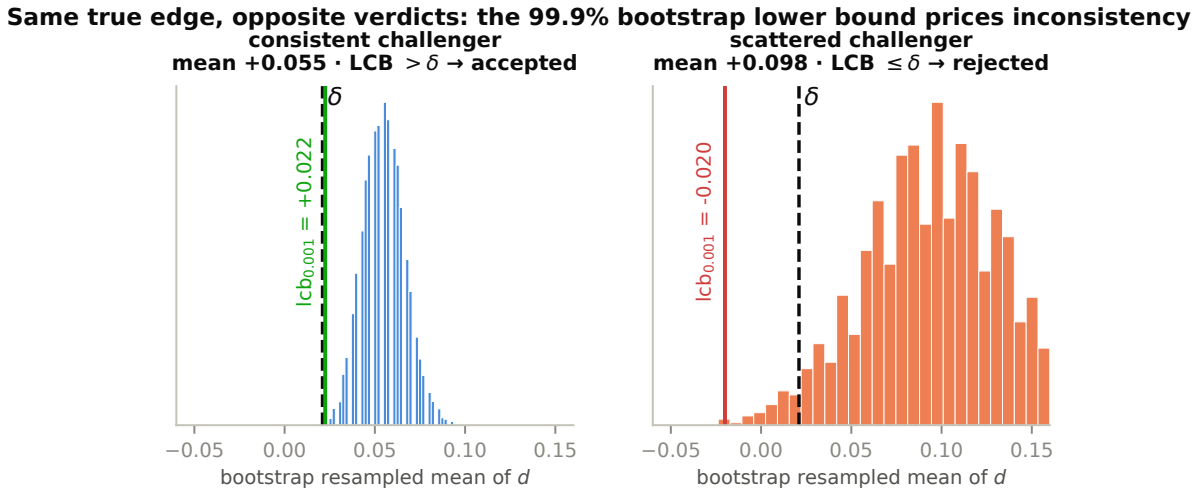

Figure 3: Bootstrap distributions of the resampled mean of d for two challengers with identical true edge (+7.5 pp) at $N = 400$, $B = 10,000$. The consistent challenger (disagreement mass 0.09) clears δ at the 0.1st percentile; the scattered one (mass 0.62) — same true edge, higher sampled mean — does not. Simulation, seed 42.

Acceptance rule. A validator accepts a challenger iff

$$\text{lcb}_{0.001}(d_{\text{pub}}) > \delta \quad \wedge \quad \hat{\mu}(d_{\text{priv}}) > 0,$$

where $\delta = \max\left(0.05 \cdot \left(1 - \text{acc}_{\text{king}}^{\text{EMA}}\right), 3\sigma_{\text{noise}}\right)$, $\text{acc}_{\text{king}}^{\text{EMA}}$ is the king's exponentially weighted public-slice accuracy (last $K = 10$ duels), and σ_{noise} is the measured king-vs-king disagreement rate.

6.2 The adaptive floor

A fixed floor is wrong at both ends of a king's life: trivial against a weak early king and mathematically unwinnable against a strong late one. Scaling the floor to the king's *remaining headroom*, $\delta = 0.05(1 - \text{acc})$, keeps the relative bar constant across the entire trajectory (Figure 4). The floor cannot be manipulated downward by miners: it moves only with the king's measured accuracy, and depressing that requires losing duels, which costs the crown.

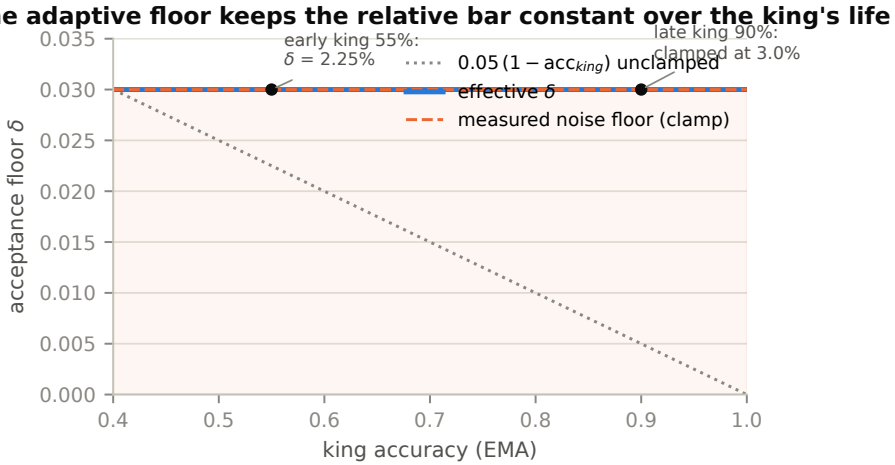

Figure 4: The acceptance floor δ as a function of king accuracy. Early kings face a proportionally meaningful bar; late kings face a fair, still-clearable one. The $3\sigma_{\text{noise}}$ clamp (orange) guarantees no verdict is ever decided inside the hardware-jitter band.

6.3 The noise clamp

Model inference is not bit-reproducible, and no available setting makes it so. Measured on the reference stack: the same prompt through the same engine, at batch of one, greedy, seeded, with `enforce_eager` on and prefix caching off, decodes differently on the second call — the fused MoE kernels reduce in a nondeterministic order, below the level any determinism flag reaches. It is not an artifact of sharding across cards; a single-GPU validator shows it too. Floating-point non-associativity flips a near-tied token and the divergence cascades through a 40-turn rollout, so a checkpoint disagrees with *itself* far more than marginally: re-scoring the same weights on the same 400 tasks flipped correctness on **84 of them (21%)**, and the paired score gap — the quantity a verdict actually turns on — carried a standard error of ~0.030 at $N = 128$.

The mechanism was built for exactly this and does not degrade under it. The duel is *paired*, so exam difficulty cancels and only d_i survives. Automated king-versus-king calibration duels run the champion against itself through the same graded path a scored duel uses, measuring σ_{noise} on that validator's own hardware in the same unit the acceptance test consumes — the standard error of the paired gap, not the per-task flip rate, since the mean is what the verdict reads and its noise shrinks with N while the flip rate does not. δ is then clamped at or above that measured floor, and acceptance is a 99.9% *lower confidence bound* rather than a point estimate, so scatter created by the noise is priced by the bootstrap instead of being assumed away. A challenger must beat the king by more than the king disagrees with itself.

Sharding is measured against the same floor rather than against zero: `scripts/gpu_equivalence.py` compares each sharded configuration to a *repeated* one-replica run, and sharded runs agree with the single-replica reference at least as well as that reference agrees with itself, so replication contributes no noise of its own. Verdicts inside the jitter band remain structurally impossible — not because the band is empty, but because it is measured and the bar sits above it.

6.4 Operating characteristics

Figure 5 plots the per-validator acceptance probability against the challenger's true edge at $N = 400$. The curve is near zero for copies and cosmetic perturbations (true edge ≈ 0), rises steeply past δ, and saturates for genuine improvements — the shape that makes copying worthless (§10) while keeping honest improvements winnable at practical sample sizes.

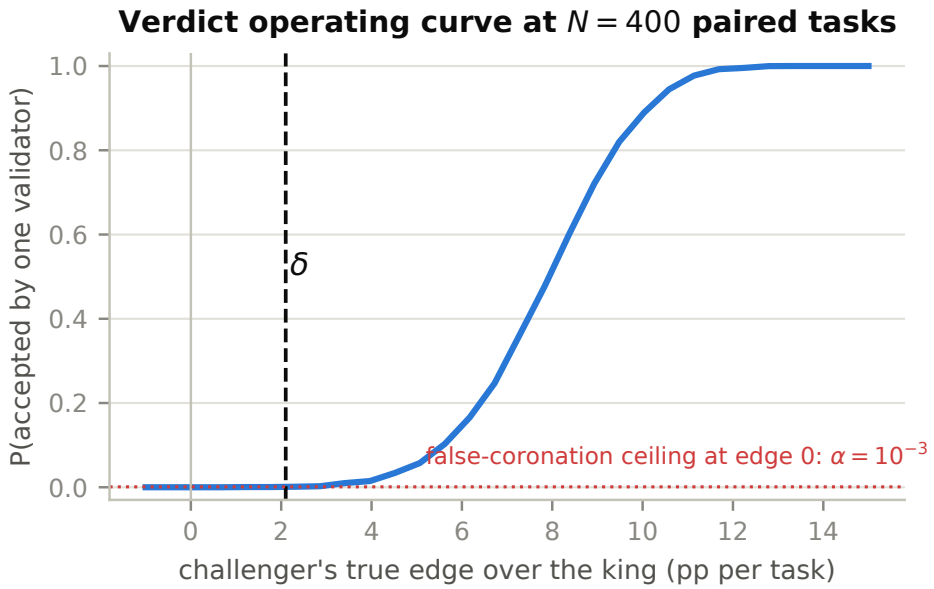

Figure 5: Verdict operating curve: probability one validator accepts, versus true challenger edge (Monte Carlo, normal approximation to the bootstrap, disagreement mass 0.30, $\delta = 0.021$). The false-coronation ceiling at true edge 0 is $\alpha = 10^{-3}$ per validator, compounded across the quorum.

7 Decentralized validation

A single evaluation server — however well audited — is an owner-shaped hole in a decentralized system: one entity controls verdicts and holds the only secret test set. Epago has no such server. Every validator runs the identical open-source engine, maintains its own private pool, and publishes its own signed verdict on-chain.

> **Coronation.** A challenger becomes king at the first block where accepting verdicts cover at least θ (default 51%) of active-evaluator stake. Coronation is a *pure function of public chain state*: every honest observer independently computes the same king at the same block. There is no coronation message, no leader, and nothing to forge.

A dissenting validator follows the quorum mechanically (its weight vector must converge) but its dissent remains permanently on record — and becomes checkable when its private pool publishes.

Weight-copying is neutralized. On Bittensor, validator rewards depend on agreement with consensus, which historically tempts validators to copy others' weight vectors instead of doing evaluation work. In Epago the correct weight vector is a deterministic function of chain state, so a copier changes nothing about which model wins: succession depends on *verdicts*, which require actually running duels. A weights-only free-rider draws dividends without contributing, but cannot corrupt selection, and is publicly identifiable (stake with no on-chain verdict trail) and excluded from quorum. Because honest validators derive the *same* vector, Yuma consensus clips essentially nothing and honest trust scores sit near 1.0, with only hour-scale coronation-race jitter, which self-heals.

8 Determinism, audit, and replay

Trust is not requested; it is checked. Every duel writes a signed **audit record** binding every input that produced the verdict: reveal block hash, derived seeds, task-set digests, the full per-task difference vector, private-pool digest and epoch, thresholds in force, and the harness version digest. The compact on-chain verdict contains this record's hash. Verdicts are then replayable at two levels:

1. **Exact (arithmetic and provenance).** Everything from the reveal block hash to the published number re-derives bit-for-bit, on a CPU, with no model and no GPU. `scripts/replay_verdict.py` recomputes both seeds from the block hash, regenerates the public task set and checks its ids digest, verifies the corpus and private-pool digests, recomputes the bootstrap LCB from the stored

difference vector, recomputes the record's `audit16`, verifies the validator's sr25519 signature, and cross-checks the on-chain verdict. Each check passes, fails, or reports itself skipped; none is a tolerance band. This level proves the validator selected the right exam, ran the mathematics honestly, and committed what it claims to have committed.

2. **Statistical (behavioral).** Re-scoring the models against the regenerated task set checks that the difference vector describes real behavior. This level cannot be exact, because inference is not (§6.3): an honest re-scoring disagrees with the original on ~21% of tasks, so it is a distributional comparison against the measured floor rather than a match. It is sufficient for what it must do — fabricating a difference vector wholesale diverges far beyond the floor and is caught, while a fabrication small enough to hide inside the floor is by construction too small to flip a verdict that cleared δ with margin.

The division of labor is deliberate. Every quantity a dishonest validator would have to falsify to steal a coronation — the seeds, the task set, the pool it committed to, the arithmetic, the signature, the chain stamp — sits in the exact level, where replay is unforgiving. The single quantity that admits only a statistical check is the one the acceptance test already treats as noisy and prices with a confidence bound over a calibrated floor.

All heavy artifacts live off-chain, content-addressed: model snapshots are pinned by a digest over their file tree, the corpus by its snapshot digest, and each is re-verified after every materialization. Validators additionally mirror the king's snapshot on decentralized S3-compatible storage, so the network survives any miner deleting their repository; because mirrors are content-addressed, a lying mirror is detected by digest mismatch rather than by trust. The public leaderboard is itself a pure export of audit artifacts — anyone can regenerate it from a validator's published state and diff it against what that validator serves. A dashboard that disagrees with the audit trail is impossible by construction.

9 Emissions and game theory

Pure winner-take-all provably collapses open competitions: rational miners hoard improvements, everyone else exits. Epago's miner-emission schedule is engineered against both failure modes (Figure 6).[1]

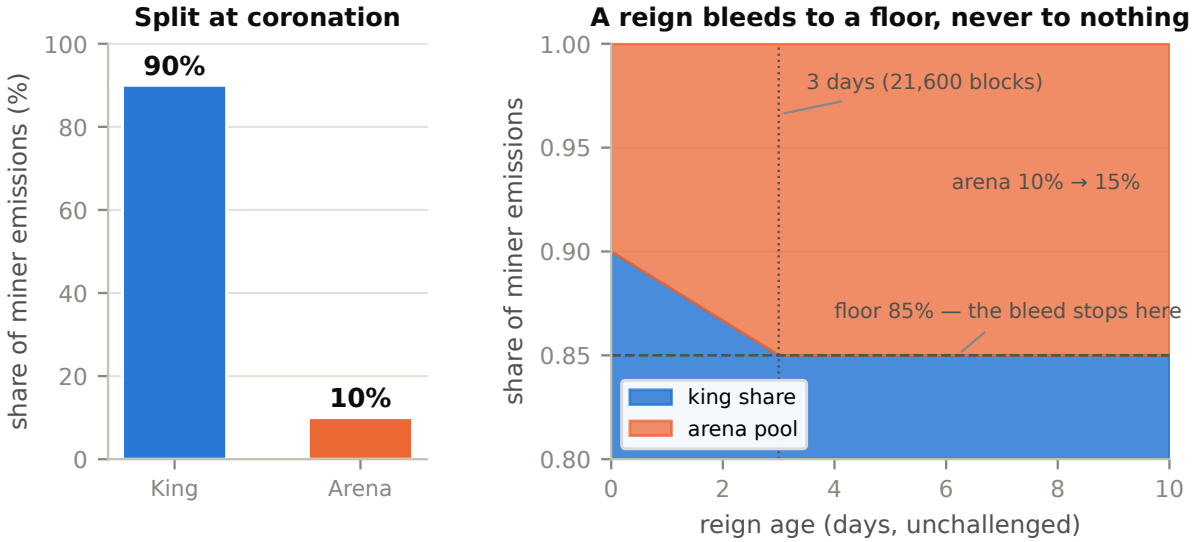

Figure 6: The miner-emission band. A reign opens at 90% of the miner slice and decays linearly to a floor of 85% over roughly three days, with the remainder flowing to an arena held by the three most recent former kings. The floor guarantees an incumbent a defined majority; the decay guarantees the crown is always worth attacking. The coronation bonus (≤3×, ∝ measured improvement, ≈24 h) makes revealing a large improvement at once dominant over dribbling it out, and never lifts the king above the top of the band.

[1]Epago is one subnet in Bittensor's dTAO market, where each subnet's share of network TAO emission is set continuously by its alpha-token price and the protocol fixes the split of that share at 41% to miners, 41% to validators, and 18% to the subnet owner [14]. The schedule described in this section — king/arena split, reign decay, coronation bonus, arena credit — governs only Epago's miner slice; it does not and cannot alter the protocol-level split.

Four terms make genuine improvement the only profitable strategy:

- **A bounded reign band.** The king's share opens at 90% and decays linearly to a floor of 85% over 21,600 blocks (~3 days), the remainder flowing to the arena. The band is deliberately narrow at both ends: an incumbent always retains a defined majority, so holding the crown is worth defending, while the crown is always worth attacking because an undefended king leaks incentive to its future challengers. Before any king exists the arena share burns rather than paying a roster that has not been earned.
- **The arena.** The three most recent former kings hold the arena, splitting its share equally, with the roster derived from the accepted verdicts on chain rather than from any validator's bookkeeping. A dethroned champion keeps earning while it prepares its next challenge.
- **Coronation bonus** ($\propto \min(\mathrm{lcb}/\delta, 3)$, paid over ~24 h, and never lifting the king above the top of the band): a large revealed improvement pays more than the same improvement dribbled out.
- **Self-dethrone inherits the reign clock**: re-crowning yourself does not reset the decay, so salami-slicing one improvement into many small coronations buys nothing.

Spam economics. Nothing is escrowed to submit — there are no bonds. Instead every resolved submission that is neither an acceptance nor a near-miss puts the author *hotkey* on an escalating cooldown (24 h base, doubling, capped ≈6 days, scaled under congestion); near-misses never cool down. The band matters: a noise-perturbed copy lands at $\mathrm{lcb} \approx 0$, so charging only decisive losses would leave exactly the copier's band free. Re-entering weights that have already dueled — under any digest — is refused at intake and cools the hotkey too. Sybil hotkeys do not help: each must win a competitive UID and pay its own registration burn, and each carries its own ladder.

10 Threat model

Each attack maps to a structural defense; the unifying principle is that **each mechanism removes exactly one way to win without building a better model**.

Attack	Structural defense
Train on the test set	Public tasks derived from a block hash that does not exist until weights are frozen (§5.1)
Overfit the open generator	Per-validator secret holdouts; private mean must be positive on every accepting validator; pools published only at rotation (§5.2)
"Trust me" evaluation	Chain-stamped, signed verdicts whose derivation replays exactly and whose scores replay against a measured noise floor (§8)
Copy or perturb the king	A copy's true edge is 0; its LCB cannot clear $\delta > 0$ — copying is *priced at zero* rather than detected, avoiding a fragile similarity-detection arms race — and a lucky draw must repeat on a fresh confirmation exam. Weights that have ever dueled are terminal under every digest (a persistent fingerprint registry), so re-uploads and re-shards are not fresh attempts (§6)
Steal a checkpoint at reveal	Digest ownership belongs to the first on-chain reveal; the timelock prevents mempool sniping; and near-ties inside the calibrated noise floor go to the *earlier* reveal, so a perturbed copy of a pending rival cannot out-place its source (§5.1)
Architecture smuggling	Config lock, safetensors-only weights (no executable code), 1.05× size cap — all enforced at intake, before any GPU work
Submission spam	Escalating hotkey cooldowns on every losing outcome; registration burn prices Sybils (§9)
A corrupt validator	Coronation requires $\geq \theta$ stake; dissent is recorded and retroactively checkable against published pools (§7)
Hardware-jitter verdicts	$\delta \geq 3\sigma_{\text{noise}}$ from continuous king-vs-king calibration (§6.3)
Silent benchmark drift	Scheduled anchoring of the king against an external public benchmark ($\approx$7 days); divergence beyond 10 pp raises a public alarm
Judge manipulation	Grading is programmatic-first; the optional LLM judge is pinned, sanitized, injection-hardened, and its invocation rate is published per duel (§5.3)

Table 3: Threat model summary. Every row is enforced by construction rather than by policy.

11 On-chain protocol

The Bittensor subtensor chain carries only fingerprints — never data. Models and the corpus live off-chain, pinned by content digests; everything on-chain is a short string in one of two facilities matched to its constraints:

- The **timelock-reveal channel** (multi-entry, chain-stamped block numbers) carries everything that must accumulate or be trustlessly ordered: challenger submissions, validator verdicts, and private-pool digest commitments. A validator cannot backdate a verdict.
- The **plaintext commitment slot** (one string per participant, 128-byte cap) carries the compact rolling audit-log checkpoint.

Wire formats are versioned (`e1` submissions, `ev1` verdicts, `ep1` pool commitments). The chain supplies exactly the three primitives it is uniquely good at: unforgeable *ordering* (who revealed first), unforgeable *timing* (no backdated verdicts), and *binding* of tiny commitments to huge off-chain objects (a 64-character digest pins a multi-gigabyte checkpoint immutably). Final rewards are set through the chain's commit-reveal weights extrinsic and allocated by native Yuma consensus every 300 blocks.

12 Implementation and deployment

The reference implementation is a "validator-in-a-box": chain client, task generator, private-pool manager, duel engine, audit log, publisher, dashboard exporter, and weight-setter in one autonomous service — corpus sync, pool rotation, calibration, anchoring, audit publication, and weight-setting all run without operator action. A validator is a machine one powers, not a job one performs. The stack ships as containers behind a single `docker compose up -d`, with images built by CI only on merges to

the main branch and updated in place by a watch service, so the fleet converges on each release without manual intervention.

Throughput. Every validator evaluates every candidate (quorum requires it), so per-validator throughput bounds verdict latency. Duels are embarrassingly parallel across tasks; the engine batches rollouts (default concurrency 16) to saturate a single evaluation GPU, with an optional low-VRAM mode and an optional remote-eval split. Demand is structurally bounded: one live submission per participant, a registration cap, escalating cooldowns, and a queue circuit-breaker that prices intake in time as latency approaches the 48-hour service target. The system delivers verdicts inside the target on one evaluation GPU and scales by adding GPUs with no mechanism change; hardware sizing is measured empirically by an included tool rather than assumed.

Intake. Submissions pass zero-GPU static checks before any rollout: config lock against the generation contract, safetensors-only weights, size cap (1.05× the king), weight-statistics sanity bounds, and a 20-task format probe (≥55% well-formed generations required) that filters checkpoints unable to operate the harness at all — so a malformed submission costs the network nothing and the submitter a cooldown.

The probe is not a formality, and its threshold is stated as a rate rather than a count: an entrant must produce well-formed episodes on at least 55% of the 20 probe tasks. The bar is deliberately below the base model's own measured compliance under the native-protocol harness — the probe exists to reject checkpoints that cannot operate the environment at all, not to gate on polish — and episode-completion discipline beyond it (finishing the research inside the turn, clock, and context budgets) remains one of the cheapest measurable improvements available to the first challengers (§1.2).

Testing. The mechanism layer is exercised by an extensive deterministic test suite (bootstrap arithmetic, seed derivations, emission schedules, cooldown escalation, quorum accounting, audit replay, storage round-trips), an adversarial mock-chain soak harness, and an external verdict-replay tool that a third party can run against any published audit record. The suite runs the eval path on a scripted backend, which is what makes its equivalence assertions meaningful: with scoring held fixed, batched rollouts match sequential ones, a low-VRAM duel matches a default one, a duel shipped to a remote eval box matches the in-process one bit for bit, and three independent validators derive bit-identical public LCBs, the same coronation block, and identical audit fingerprints. Those are claims about the protocol, which is deterministic; none of them asserts that the GPU engine is, and the mechanism is built so that none of them needs to be.

13 The network: the layered architecture

No mechanism in §4 through §11 depends on the domain. The selection machinery handles exactly one object — the vector d of per-task correctness differences — and is indifferent to what produced each entry. Everything domain-specific lives in the **generation contract**: a corpus snapshot digest, a set of task templates, the tool interface, and the architecture pins. That separation is what makes the layers below expansions of configuration and construction rather than rewrites of the protocol, and each advances independently of the others.

This section states the stack of Table 2 in full, layer by layer, with the honest status of each. A paper that condemns unverifiable claims cannot make unverifiable claims about itself, so this section and the roadmap of §16 are where the paper marks its own registers explicitly — and they are the only two places it does so. There are exactly two marks and there is no third: RUNNING TODAY for capability that is built, measured, and specified here against the genesis-generation parameters of Appendix A, where every number is either a published result attributed to its source or an Epago measurement labelled as such (§1.2); and THE HORIZON for architecture and roadmap. Everywhere else this paper simply says what runs and what is planned. One rule follows from that discipline and governs the whole stack: **the acceptance test of §6, the quorum rule of §7, and the audit architecture of §8 are fixed points.** No layer described here proposes a new consensus, a new scoring rule, a new judge, or a new emission scheme. Where a layer would require touching one of those fixed points, that is stated in the layer's own subsection and the layer stays on the horizon until it is resolved.

RUNNING TODAY

§13.1 and §13.2 describe capability that exists. L0 is specified in full by §3 through §12 of this paper and is running. L1 has one live instance — the genesis generation — and a documented configuration path to the next ones that requires no protocol change: a new domain is a contract to write, not a codebase to build.

13.1 L0 — the provable-improvement engine

L0 is the whole of §3 through §12, restated once as a layer. Open models compete in paired duels on an exam that cannot be gamed: generated after the competing models are frozen (§5.1), half of it drawn from per-validator private holdouts minted from documents published after training and republished in full at rotation (§5.2), graded mechanically against keys proven to exist in a pinned corpus (§5.3), decided by a 99.9% bootstrap lower confidence bound against an adaptive, noise-clamped floor (§6), coronated at the first block where accepting verdicts cover a stake quorum with no coronation authority anywhere in the loop (§7), and replayable by any stranger from public chain state — exactly for the protocol, statistically for the scores, and specified either way (§8). A challenger takes the crown only by proving an improvement.

Two properties of L0 are what every layer above inherits, and neither is negotiable. First, **there is no trusted judge**: correctness is a comparison against a key that was proven against the corpus before the task was posed — never an opinion, never a vote among participants, never a model asked what it thinks (§5.3). Second, **every verdict is replayable**: a stranger with public data and a CPU can recompute the seeds, the exam, the arithmetic, the digest and the signature exactly, and a stranger with one GPU can check the behavior within a measured noise budget — ~21% per-task self-disagreement, gap SE ~0.030 at $N = 128$ (§6.3, §8). Any layer that could not be built without surrendering one of these is a layer this network does not build.

13.2 L1 — specialized research intelligence

A domain is a chain generation, and a chain generation is a contract. The genesis generation is the one live instance; it is an instantiation of the task class of §2, not the definition of the product. Miners specialize, each domain carries its own crown and its own market, and all of them run over one shared mechanism. The three dials of a generation contract are domain, task rung, and modality.

Domains as chain generations. Instantiating law requires a corpus snapshot (case law, statutes, filed contracts), task templates whose answers are verifiable against that snapshot, and the same architecture pins; it requires no change to duel scoring, exam derivation, acceptance statistics, coronation, audit, or emissions. Generations therefore run in **parallel** — each with its own king, its own private-pool feed, and its own reign accounting — over one shared mechanism. Because they are parallel rather than sequential, there is no ordering to declare: a generation exists when a corpus satisfies the two invariants below and someone writes the contract for it.

Two invariants must hold for a candidate domain, and they are the real admission criteria. First, **answers must be checkable against the corpus without an opinion**: the grading cascade of §5.3 is programmatic first, and a domain whose questions can only be graded by a judge model would reintroduce exactly the trusted party the protocol removes. Second, **fresh documents must arrive on a schedule**, because the private-pool tripwire of §5.2 is only a tripwire if it can be minted from material published after a challenger's training cutoff. Law, patents, and regulatory filings satisfy both by construction: they are dated, public, citable, and continuously issued. Domains that satisfy neither — open-ended advice, matters of taste — are outside the protocol's reach and are not claimed.

The task ladder. Extraction is rung one, chosen because its answer keys are the cleanest. The ladder above it, in an evidence-synthesis generation, is the remainder of the review workflow: **screening** a candidate set against eligibility criteria, **cross-study synthesis** over extracted effect estimates, **risk-of-bias appraisal**, and finally **drafted review sections** with every sentence attributed. Each rung

is worth more per unit of work than the one below it, and each is a task-template change inside an existing generation rather than a new mechanism.

Each rung also costs grading determinism, and the protocol's response is identical at every rung: move the subjectivity into the answer key, never into the judge. A screening task is scored as set agreement against a pre-proven inclusion list; a synthesis task is scored on the extracted numbers and on the direction and magnitude of the pooled effect, not on prose quality; an appraisal task is scored against published domain-level judgments. Where a rung genuinely cannot be reduced to a mechanical key, it does not enter the exam — and the per-duel judge-invocation rate published under §5.3 is the public instrument that keeps that promise checkable rather than declared.

Modality. Most reported quantitative results are not in prose. They are in tables — arm sizes, event counts, effect estimates with confidence intervals, subgroup rows — and in figures such as survival curves and forest plots. This is where every current model is weakest, which is precisely why it is the most defensible capability frontier for a competition that pays only for measured improvement: there is more headroom here than anywhere in the text pipeline, and the terrain is not yet occupied.

Table and figure extraction also happens to suit Epago's machinery unusually well, because it is brutally checkable: a number in a table cell either matches the key or it does not, so this dial adds capability without adding grading ambiguity. Tables are a corpus-and-template change within the current architecture (store the table structure, template questions over its cells); figures additionally require a multimodal architecture pin, which is a new generation rather than a new protocol.

THE HORIZON

Everything from here to the end of the section is architecture and roadmap. None of it is live, none of it earns emissions, and none of it is offered as capability the network has. It is stated plainly rather than tentatively, because each layer is a construction problem over an engine that already works — but it is marked plainly, because a paper that asks the reader to distrust unverifiable claims has to tell the reader which of its own are not yet verifiable. The fixed points of §6, §7 and §8 hold throughout; where a layer presses against one, the subsection says which and says how far.

13.3 L2 — verification

The largest single failure mode in AI research output is fabricated provenance — and it is the one failure mode that is mechanically checkable. The figures are in §2.2 and they are not marginal: in one audit of 20 medical questions, 69% of the references ChatGPT supplied did not exist while 95% named real authors with genuine related publications [1], and an audit of GPT-4o across six literature reviews found 19.9% of its citations wholly invented and 45.4% of the genuine ones erroneous, with 64% of the fabricated DOIs resolving to real but unrelated papers [17]. A network that could reliably answer *does this source say what it is claimed to say?* would be addressing the defect that most disqualifies generated research from professional use.

That question is checkable without a judge. Given a pinned corpus, a claim, and a citation, establishing that the cited document exists, that it reports the quantity attributed to it, and that the attributed value matches within tolerance is a comparison against a document — the same shape of check the grading cascade of §5.3 already performs on extraction answers. Citation and provenance verification is therefore a legitimate new **task family** for this protocol rather than a new protocol: tasks whose keys are proven against the corpus at generation time, difficulty-filtered into the same solve band (§3), scored per task into the same difference vector d, and decided by the same acceptance test on the same quorum rule. Nothing in §6, §7 or §8 moves.

The boundary is deliberate, and it is where the honesty of this layer lives. In scope: does the cited document exist, does it contain the claimed value, does the reported figure match the source. Out of scope: is this claim overstated, is this study well designed, is this synthesis fair. Not because

those questions do not matter — they are among the most valuable questions in evidence work — but because answering them requires a trusted judge, and Epago grades without one (§5.3, §13.1). A verification layer that admitted subjective claims would have to admit either a judge model or a vote among participants, and both are precisely the trust this protocol exists to remove. L2 is scoped to the mechanizable core and stops there, deliberately and permanently.

Status: design stage. The mechanism it would run on is built and unchanged. What does not yet exist is the task-template family, the key-generation procedure that proves a provenance key against the corpus as cleanly as an extraction key is proven today, and the calibration that keeps such tasks inside the measured difficulty band where a label carries signal. Until those exist, L2 is a design, and this paper claims nothing more for it.

13.4 L3 — the living knowledge layer

Research is thrown away. An agent decomposes a question, searches, opens forty documents, reconciles three that disagree, and emits a short answer; the evidence trail and the reconciliation are discarded, and the next question over the same literature starts from zero. The waste is structural rather than accidental — an answer is the wrong container for what the work produced.

The protocol already emits the raw material for the alternative. Every duel yields verified `(task, answer, verdict)` records over documents newer than the models under test, published against pre-committed digests on a fixed schedule (§8, §14). What is missing is not the data but the **shape**. A record structured as *claim → evidence → source → confidence → timestamp → relationships* accumulates in a way that an answer string cannot: stored that way, verified results compound into a knowledge graph in which what is known, what it rests on, when it was established, and what it contradicts are all first-class and all traceable to a specific document in a pinned corpus. The value is compounding rather than additive — each new verified claim is also a new edge against everything already in the graph.

What this layer may not do. It may not establish truth by aggregation, by popularity, or by any participant's assertion. A claim enters because a mechanically graded duel over a pinned document established it, and it carries the digest of the duel that did so; anything a stranger cannot re-derive from published artifacts does not belong in the graph. The graph is a record of what was verified, never a vote on what is true.

Status: roadmap. Construction, storage, query semantics, and conflict handling are unbuilt. The input stream that would feed them is running today.

13.5 L4 — continuous research

A report is an answer at a timestamp. Evidence keeps arriving; the report does not keep up. Where that gap has been measured, it is wide: 64.3% of Cochrane reviews are never updated, at a median of 57 months between updates.[2] Standing research objectives that re-run themselves when new evidence publishes, and report what changed and what it changed from, are a real and unserved need in exactly the kind of workflow §2.1 describes.

The component usually hardest to build for that is the one Epago already runs. The private-pool feed is a bounded, automated, revision-pinned stream of newly indexed papers, rotating on a fixed schedule and minted into tasks whose answers postdate the models that answer them (§5.2). Today it exists to be a contamination tripwire. The same stream pointed at a standing question rather than at a fresh exam is the mechanism of a living review.

Corpus reach, and its mechanism cost. The pinned corpus exists for exactly one reason: a byte-identical world is what lets every validator pose the identical exam, which is the one half of reproducibility that can be made exact (§3, §8). It is not the deployment target. The reach ladder is **pinned snapshot → live retrieval** against a public index → **the customer's own private document store**, and the learned behavior transfers across all three because the agent only ever

[2] Cochrane review output and update-lag figures are drawn from the published Cochrane Database of Systematic Reviews literature (PMC12362767, 2025; PMC11795965). They are external figures about the review ecosystem, not Epago measurements.

sees two tool signatures, `Search` and `Browse`. What sits behind them is plumbing; a model selected for procedural discipline over a pinned corpus is retriever-agnostic by construction, and swapping the backend is an integration task, not a retraining run.

The mechanism cost of climbing that ladder is explicit. Live retrieval breaks the byte-identical world that makes the *exam* exactly reproducible and so raises the noise on d above the engine floor of §6.3, so a duel over a live index must either snapshot the index for the duration of the duel — restoring an identical exam at the price of freshness *within* a duel — or accept a wider disagreement budget, which the king-versus-king calibration already measures rather than assumes. Private customer stores are a deployment mode and never an evaluation mode: nothing scored on data a validator cannot see could be replayed by a stranger, and a verdict that cannot be replayed is precisely what this protocol refuses to issue. Evaluation stays pinned; deployment goes wherever the documents are.

Status: roadmap, and the one layer that presses on a fixed point. Standing objectives over a live index would break the byte-identical world that makes the exam behind d exactly reproducible. The resolution is stated above and it is not a change to the acceptance test: it is a choice between snapshotting the index for the duration of a duel and accepting a wider disagreement budget that the king-versus-king calibration of §6.3 already measures. Because that choice has not been made and its operating characteristics have not been measured, L4 stays here rather than migrating into the running register. Evaluation stays pinned; a live index is a deployment question until the day it is a measured one.

13.6 L5 — machine-to-machine intelligence

The largest consumer of research will not be a person. It will be another agent: a trading agent that needs the macroeconomic evidence behind a position, a coding agent that needs the current behavior of an API rather than the behavior frozen into its training data, a biotech agent that needs findings published last month. Those callers do not want prose. They want a machine-readable answer with its evidence attached — the claim, the source, the date, and a trail the caller can check without trusting the responder.

That is the same object L0 already produces per task and the same object L3 would accumulate; L5 is the interface to it. Verified, cited, machine-readable research is served through an API, with the model that produced it pinned by content digest and the selection procedure behind that model replayable by the caller from public chain state (§8). The positioning follows directly and is worth stating: Epago is **intelligence infrastructure for AI agents**, not a consumer research product. The regulated, reproducibility-bound human users of §2.2 want the same property for a different reason — under the 2025 RAISE guidance the model version is part of the reportable method [27], and a digest-pinned answer satisfies that where a silently updated endpoint cannot.

Status: roadmap. One boundary does not move at this layer and is worth making explicit, because it is where a demand-side network would normally start corrupting its own evaluation: **nothing served through the API is ever an input to a verdict.** Usage, customer satisfaction, and downstream agent behavior are not signals the protocol reads. Selection is decided by mechanically graded duels on freshly generated exams and by nothing else, exactly as specified in §5 through §7, whether the network serves one caller or a billion.

14 Verified reward data as a first-class output

The duels produce a king. They also produce something the protocol was not primarily designed to produce, and which may prove the more durable asset: a continuously growing archive of **verified reward data**.

Every rollout in every duel yields a record — the task, the committed answer, and a mechanically determined correct/incorrect label — and the task set, the model digests, the harness version, and the pinned decoding parameters are published alongside it (§8), so the full reasoning-and-tool trajectory behind each label is regenerable by anyone rather than merely asserted. At the genesis defaults that is $N \approx 1,000$ paired tasks, roughly 2,000 graded episodes, per duel per validator; a single validator

sustaining one duel per day accumulates on the order of 7×10^5 labelled episodes per year, and the fleet multiplies it. The labels are neither human annotations nor model opinions: they are comparisons against keys proven to exist in the corpus at generation time (§5.3).

This is the input that post-training is short of. The leverage in frontier reasoning models has moved into the post-training stage [29], and the binding constraint on scaling reinforcement learning there is not compute but environments whose rewards resist hacking — a market already exceeding $1B per year, in which verification rather than task authoring is the leading bottleneck [30]. Epago's exam is a verification mechanism first and a competition second; the reward data is what falls out of running it at scale.

Three properties make the archive hard to replicate. It is **fresh**: private-slice tasks are minted from documents published after the models under test were trained, so it cannot be reconstructed from any existing scrape (§5.2). It is **adversarially calibrated**: the task families are constructed — and re-validated by standing ablation whenever they change (§5.4) — so that the reigning king solves them sometimes but not always, the regime in which a label carries the most signal and the least redundancy; and because each release is re-measured against the current king, the archive stays at the frontier instead of saturating. And it is **time-locked**: the records are produced by a schedule of duels against successive kings, so a competitor cannot buy the history, only start their own clock.

The public part of it is a public good by construction. Private pools publish in full at rotation (≈6 days), and public task sets publish on the audit delay (≈7 days), each against a digest committed before it was used. The archive is therefore not a proprietary hoard but a continuously released, cryptographically dated record that anyone can train on and anyone can check — which is the same property that makes the verdicts trustworthy, applied to the data instead of the decision.

The result is a closed loop: fresh documents mint tasks, tasks drive duels, duels emit verified labels, labels train better challengers, better challengers make the crown more valuable, and a more valuable crown attracts the miners who run the next round of duels. Every turn of the loop leaves behind an asset that does not change hands when the reign does.

The flywheel is what makes the layers of §13 reachable, and it is why they are construction rather than invention. Each layer consumes some part of what the loop already produces. L1 multiplies the loop rather than modifying it: every generation runs its own, over its own corpus, with its own king and its own fresh-document feed. L2 adds a task family whose labels the same loop produces, on the same exam machinery, against the same acceptance test. L3 is the loop's output **kept** instead of discarded — the same verified records, structured to accumulate. L4 is the loop's fresh-document feed pointed at standing questions instead of one-shot exams. L5 is the loop's verified output addressed to a caller. None of that changes the loop; all of it is downstream of running it. And running it is the part a competitor cannot shortcut: because the archive is fresh, adversarially calibrated, and time-locked, acquiring the asset would mean operating the machine for a year, which is the same thing as building the network.

15 Related work

Contamination-resistant evaluation. LiveBench [4] and LiveCodeBench [5] defend against test-set leakage by continuously introducing tasks that postdate model training; LiveBench additionally refreshes roughly a sixth of its questions each month and grades every answer against a ground-truth key rather than with an LLM judge [4]. A parallel line defends by secrecy instead: MMLU-CF keeps its test split closed, on the finding that contamination otherwise inflates results on any fully open benchmark [6]. Epago generalizes both defenses. It replaces the calendar with a block hash — tasks are generated from entropy that provably does not exist until after weights freeze, which yields per-submission freshness with permanent replayability, a property calendar-based benchmarks cannot offer — and it replaces the single closed test set with per-validator secret holdouts published in full at rotation, removing the trusted operator that a closed-source benchmark still requires. Its grading follows the same programmatic-first discipline, with the LLM judge disabled by default and its invocation rate published per duel.

Agentic research agents and benchmarks. Agentic research capability is measured by GAIA [31], BrowseComp [13], WebWalkerQA [32], FRAMES [11], xbench-DeepSearch [12], and Humanity's Last Exam [10]; open-source systems now match or exceed closed deep-research products on several of these [2]. Epago's genesis king is Tongyi-DeepResearch-30B-A3B [9], whose reported agentic-search results — GAIA 70.9, BrowseComp 43.4, BrowseComp-ZH 46.7, WebWalkerQA 72.2, xbench-DeepSearch 75.0, FRAMES 90.6, and Humanity's Last Exam 32.9 — are the capability floor the competition starts from, not Epago's own claim (Table 1). Those figures exceed OpenAI o3, DeepSeek-V3.1, and Claude-4-Sonnet on five of seven of these benchmarks — Humanity's Last Exam 32.9 against 24.9 / 29.8 / 20.3, FRAMES 90.6 against 84.0 / 83.7 / 80.7, xbench-DeepSearch 75.0 against 67.0 / 71.0 / 65.0 — while proprietary o3 still leads on BrowseComp, 49.7 against 43.4 [9]. The genesis king therefore starts ahead of the closed frontier on most agentic-search axes, at ~3.3B active parameters per token against a 671B-parameter comparator, and behind it on at least one: that remaining gap is headroom the competition exists to close. What the comparison supports is a claim about grounded research per unit of compute, not about general capability. Epago's contribution is not the model but the selection mechanism around it.

Decentralized and incentivized training. Distributed low-communication training has scaled from 10B [33] to 32B with verified distributed RL [34], demonstrating that gradient production itself no longer requires a single datacenter. The frontier is now decentralizing the environment layer itself: INTELLECT-3, a 106B-parameter mixture of experts, reached frontier-competitive reasoning by reinforcement-learning post-training a mid-size base on community-contributed evaluation environments, matching a model with more than three times its parameter count [35]. This locates the leverage in post-training and evaluation — the stage that consumed roughly 20% of DeepSeek-R1′s base-model pretraining compute by FLOP, and far less than that for several other reasoning models [29] — which is also the stage a single-GPU Epago participant operates in, and the stage whose leading bottleneck is verification that resists reward hacking [30]. The gap this leaves is instructive: INTELLECT-3′s own post-training ran on a centralized 512-GPU cluster [35], so the environment and selection layer remains centralized even at the decentralized-training frontier. These efforts decentralize *gradient production*; Epago decentralizes *model selection* — the two are complementary, and Epago is agnostic to how a challenger was trained. Prize- and leaderboard-driven development (the Netflix Prize, Kaggle, the Stockfish/Fishtest pipeline, SWE-bench) demonstrates that open continuous competition finds state of the art quickly; Epago contributes the missing piece for open-weights competition: an exam that cannot be gamed and verdicts that need no referee.

Verifiable randomness. The commit-reveal timelock and block-hash seeding follow the public-randomness literature; threshold beacons such as drand [36] provide unbiasable post-commitment entropy, and the implementation includes a beacon client as an upgrade path for public-task seeding beyond block hashes.

16 Roadmap

Epago is organized around **chain generations**. Each generation pins a base architecture, corpus snapshot, environment, and parameter set in a single contract; the competition runs until the community advances the contract. The genesis generation fields Tongyi-DeepResearch-30B-A3B (30.5B parameters, ~3.3B active per token) over a pinned scientific-paper corpus, and names its chain of improved descendants **EPAGO-DR-30B**. Because model, corpus, and environment are configuration rather than code, future generations — larger or multimodal bases, refreshed or widened corpora, new task families, beacon-seeded public tasks — are contract changes, not rewrites.

RUNNING TODAY

The near-term path runs from multi-validator testnet operation (live quorum, real GPU timing, chain-behavior soak) to mainnet launch with the full economic schedule active. The first objectives inside the genesis generation are the measured deficits of §1.2: episode completion — the telemetry shows most lost points sit in research episodes abandoned at the turn, clock, or context budget

rather than answered wrongly — and the gap between full-tooling accuracy (~22%) and the oracle-retrieval ceiling (91.0%), which is precisely the retrieval skill the exam prices. Inside the genesis generation, expansion then proceeds up the task ladder and into table modality (§13.2). Throughout, the archive of §14 accumulates as a published, dated by-product of the competition rather than as a separate programme.

THE HORIZON

Beyond the genesis generation, the path is the stack of §13, and it is a direction of travel rather than a schedule: parallel L1 generations in further domains — case law and contracts, patents, regulatory filings — each admitted only on the two invariants of §13.2 — answers mechanically checkable against the corpus, documents arriving on a schedule; an L2 verification task family once provenance keys can be proven against a corpus as cleanly as extraction keys are today (§13.3); and L3 through L5 — the knowledge layer, continuous research, and the agent-facing interface — which are architecture today and are marked as such wherever they appear. No item on that list is permitted to alter §5 through §9. A proposal that required altering them would be a different protocol, and would have to be published as one.

17 Conclusion

Deep research is a procedure, and procedures fit in small models. That is why an open checkpoint activating ~3.3B parameters per token already leads a 671B one on most agentic deep-research benchmarks, and trails it on at least one — and it is also why that frontier stops moving the moment its authors move on. What is missing is not capability but a mechanism: something that makes the next step worth taking and proves it was real.

Epago is that mechanism. It turns "which model is better?" from a claim into a checkable statement, re-proved publicly every time a challenger is submitted. Generating the exam after weights freeze removes training on the test; distributed secret holdouts with delayed publication remove generator overfitting without asking for trust; the bootstrap bound over an adaptive, noise-clamped floor removes luck, copies, and jitter; stake-quorum coronation removes the trusted operator; and the audit architecture makes every one of these removals verifiable by a stranger from public data. What is left is deliberately narrow and deliberately portable: a domain is a contract, the ladder above extraction is a template change, and the verified reward data the machine emits along the way is scarcer than the crown it awards.

That mechanism is a foundation, not a destination. Search engines find information; language models generate answers; neither establishes that an answer is true, that it came from somewhere, or that this month's system is better than last month's. A network that grades mechanically, publishes its evidence, and pays only for improvements a stranger can re-derive is the missing third thing — and once it exists, what sits above it is construction rather than new trust: domains with their own crowns and their own markets (§13.2), verification of the provenance that generated research most reliably fabricates (§13.3), a knowledge layer that keeps what is presently discarded (§13.4), research that maintains itself as evidence arrives (§13.5), and an interface that serves all of it to the agents that will be its largest consumers (§13.6). This paper states that boundary plainly wherever it appears, and marks it explicitly where the layers are set out (§13) and scheduled (§16). L0 and the genesis generation are running; the rest is stated as architecture, and stated with confidence, because the discipline that makes the first layer checkable is the discipline that governs every layer above it — and because a vision worth having is one whose author will tell you which parts of it are not built yet.

What survives is the only strategy the protocol is designed to reward: building a genuinely better open deep-research model, and then doing it again.

Epago is open-source under the MIT license. This whitepaper describes the protocol as implemented; parameters cited are genesis-generation defaults and are configurable per chain generation. Material marked THE HORIZON describes architecture and roadmap rather than deployed capability, and nothing so marked is live or earning emissions. Nothing herein is an offer of securities or financial advice.

18 Appendix A: Genesis-generation parameters

Parameter	Default	Role
N_PUB_TASKS / N_PRIV_TASKS	800 / 200	public / private slice sizes per duel
BOOTSTRAP_B	10,000	bootstrap resamples
EVAL_ALPHA	0.001	one-sided LCB level (99.9%)
DELTA_C	0.05	floor coefficient: $\delta = c(1 - \text{acc}_{\text{king}})$
DELTA_NOISE_MULTIPLIER	1.0	noise clamp multiplier
KING_ACC_EMA_K	10	EMA window (duels) for king accuracy
TASKGEN_RELEASE	POOL1	sealed-pool intersection family; one-, two- and three-hop tiers at 35/35/30
ROLLOUT_MAX_TURNS	40	agentic turns per task
ROLLOUT_CONTEXT_TOKENS	32,768	rollout context budget
ROLLOUT_TEMPERATURE / seed	0.0 / 42	greedy decoding, pinned
ROLLOUT_REPETITION_PENALTY	1.05	pinned; breaks greedy repetition loops
ROLLOUT_TIMEOUT_S	3,600	wall clock per rollout
ROLLOUT_CONCURRENCY	32	batched rollouts per GPU
BLOCKS_UNTIL_REVEAL	5	submission timelock
MAX_CHALLENGER_SIZE_RATIO	1.05	size cap vs king
PRIVATE_POOL_ROTATION	43,200 blocks	≈6 days; pool publishes at rotation
FORMAT_PROBE_MIN_COMPLIANCE	0.55	intake probe: minimum well-formed-episode rate over 20 probe tasks
SLA_TARGET_HOURS	48	verdict service target
QUEUE_BREAKER_HOURS	36	intake circuit-breaker threshold
ARENA_MAX_KINGS	3	former kings sharing the arena equally
NEAR_MISS_RETRIES	1	free retry on fresh exam
WEIGHT_INTERVAL_BLOCKS	300	weight-setting cadence
king / arena share	0.90 → 0.85	miner-emission band; remainder to the arena
REIGN_DECAY_BLOCKS	21,600 blocks	≈3 days, linear 90% → 85% floor
CORONATION_BONUS	cap 3×, 7,200 blocks	∝ min(lcb/δ, cap), ≈24 h
ANCHOR_INTERVAL	50,400 blocks	≈7 days external anchor; also fires at every coronation
ANCHOR_DIVERGENCE_ALERT	0.10	drift alarm threshold
AUDIT_PUBLISH_DELAY	50,400 blocks	≈7 days public-task release
CROSS_GPU_NOISE_BUDGET	0.03	max cross-hardware disagreement rate in testing

References

[1] J. Gravel, M. D'Amours-Gravel, and E. Osmanlliu, "Learning to Fake It: Limited Responses and Fabricated References Provided by ChatGPT for Medical Questions," *Mayo Clinic Proceedings: Digital Health*, 2023, doi: 10.1016/j.mcpdig.2023.05.004.

[2] MiroMind Team, "Open-Source Agentic Deep Research at the Frontier," 2026.

[3] H. Zhang and et al., "A Careful Examination of Large Language Model Performance on Grade School Arithmetic," 2024. [Online]. Available: https://arxiv.org/abs/2405.00332

[4] C. White and et al., "LiveBench: A Challenging, Contamination-Free LLM Benchmark," 2024.

[5] N. Jain and et al., "LiveCodeBench: Holistic and Contamination Free Evaluation of Large Language Models for Code," 2024.

[6] Q. Zhao and et al., "MMLU-CF: A Contamination-free Multi-task Language Understanding Benchmark," 2024. [Online]. Available: https://arxiv.org/abs/2412.15194

[7] S. Singh and et al., "The Leaderboard Illusion," 2025. [Online]. Available: https://arxiv.org/abs/2504.20879

[8] Epoch AI, "Frontier Labs' Share of Global AI Compute." [Online]. Available: https://epoch.ai/gradient-updates/frontier-labs-dont-use-most-ai-compute

[9] Tongyi DeepResearch Team, "Tongyi DeepResearch Technical Report," 2025. [Online]. Available: https://arxiv.org/abs/2510.24701

[10] L. Phan and et al., "Humanity's Last Exam," 2025.

[11] S. Krishna and et al., "Fact, Fetch, and Reason: A Unified Evaluation of Retrieval-Augmented Generation," 2024.

[12] xbench Team, "xbench: Tracking Agents Productivity Scaling with Profession-Aligned Real-World Evaluations," 2025.

[13] J. Wei and et al., "BrowseComp: A Simple Yet Challenging Benchmark for Browsing Agents," 2025.

[14] Bittensor, "Emissions," Bittensor Documentation. [Online]. Available: https://www.bittensor.com/docs/concepts/emissions

[15] M. J. Page and et al., "The PRISMA 2020 Statement: An Updated Guideline for Reporting Systematic Reviews," *BMJ*, 2021, doi: 10.1136/bmj.n71.

[16] W. H. Walters and E. I. Wilder, "Fabrication and Errors in the Bibliographic Citations Generated by ChatGPT," *Scientific Reports*, 2023, doi: 10.1038/s41598-023-41032-5.

[17] J. Linardon and et al., "Influence of Topic Familiarity and Prompt Specificity on Citation Fabrication in Mental Health Research Using Large Language Models: Experimental Study," *JMIR Mental Health*, 2025, doi: 10.2196/80371.

[18] Y. Kim and et al., "Medical Hallucinations in Foundation Models and Their Impact on Healthcare," 2025. [Online]. Available: https://arxiv.org/abs/2503.05777

[19] G. Perkins, N. W. Anderson, and N. C. Spies, "Retrieval-Augmented Generation Salvages Poor Performance from Large Language Models in Answering Microbiology-Specific Multiple-Choice Questions," *Journal of Clinical Microbiology*, 2025, doi: 10.1128/jcm.01624-24.

[20] F. Hoffmann and et al., "Nearly 80 Systematic Reviews Were Published Each Day: Observational Study on Trends in Epidemiology and Reporting Over the Years 2000-2019," *Journal of Clinical Epidemiology*, 2021, doi: 10.1016/j.jclinepi.2021.05.022.

[21] M. Michelson and K. Reuter, "The Significant Cost of Systematic Reviews and Meta-Analyses: A Call for Greater Involvement of Machine Learning to Assess the Promise of Clinical Trials," *Contemporary Clinical Trials Communications*, 2019, doi: 10.1016/j.conctc.2019.100443.

[22] R. Borah, A. W. Brown, P. L. Capers, and K. A. Kaiser, "Analysis of the Time and Workers Needed to Conduct Systematic Reviews of Medical Interventions Using Data from the PROSPERO Registry," *BMJ Open*, 2017, doi: 10.1136/bmjopen-2016-012545.

[23] The HIPAA Journal, "What Are the Penalties for HIPAA Violations?." [Online]. Available: https://www.hipaajournal.com/what-are-the-penalties-for-hipaa-violations-7096/

[24] European Union, "Art. 9 GDPR — Processing of Special Categories of Personal Data," General Data Protection Regulation (Regulation (EU) 2016/679). [Online]. Available: https://gdpr-info.eu/art-9-gdpr/

[25] R. F. Qayyum and M. F. Sattar, "Summary of EDPB's Opinion 28/2024 Concerning AI Models & Processing of Personal Data," Securiti. [Online]. Available: https://securiti.ai/summary-of-edpb-opinion-282024-concerning-ai-models-processing-of-personal-data/

[26] European Union, "Article 99: Penalties," EU Artificial Intelligence Act (Regulation (EU) 2024/1689). [Online]. Available: https://artificialintelligenceact.eu/article/99/

[27] E. Flemyng and et al., "Position Statement on Artificial Intelligence (AI) Use in Evidence Synthesis Across Cochrane, the Campbell Collaboration, JBI and the Collaboration for Environmental Evidence 2025," *Cochrane Database of Systematic Reviews*, 2025, doi: 10.1002/14651858.ED000178.

[28] Z. Yan and et al., "LiveMedBench: A Contamination-Free Medical Benchmark for LLMs with Automated Rubric Evaluation," 2026. [Online]. Available: https://arxiv.org/abs/2602.10367

[29] J. You, "How Far Can Reasoning Models Scale?," Epoch AI Gradient Updates. [Online]. Available: https://epoch.ai/gradient-updates/how-far-can-reasoning-models-scale

[30] J.-S. Denain and C. Barber, "An FAQ on Reinforcement Learning Environments," Epoch AI Gradient Updates. [Online]. Available: https://epoch.ai/gradient-updates/state-of-rl-envs

[31] G. Mialon and et al., "GAIA: A Benchmark for General AI Assistants," 2023.

[32] J. Wu and et al., "WebWalker: Benchmarking LLMs in Web Traversal," 2025.

[33] Prime Intellect Team, "INTELLECT-1: The First Globally Trained 10B Parameter Model," 2024. [Online]. Available: https://www.primeintellect.ai/blog/intellect-1-release

[34] Prime Intellect Team, "INTELLECT-2: A Reasoning Model Trained Through Globally Decentralized Reinforcement Learning," 2025.

[35] Prime Intellect Team, "INTELLECT-3: Technical Report," 2025. [Online]. Available: https://arxiv.org/abs/2512.16144

[36] E. Syta and et al., "Scalable Bias-Resistant Distributed Randomness," in *IEEE Symposium on Security and Privacy*, 2017.