The Courtroom Challenge

Can Any Open-World AI Thrive Without Paying the Validation Bill?

This note poses a conjecture, not a leaderboard: automate validation, prove it's unnecessary, or price human judgment honestly.

John Repsys*

May 4, 2025

Abstract

Large language models hallucinate; ad-hoc patches conceal the true cost of verification. We therefore pose the Courtroom Challenge: Conjecture 1. For any unbounded open-world text stream, a system that lacks both (i) a persistent precedent store and (ii) a friction-priced validation loop will accumulate unbounded prequential regret. We formalise the state machine, cost ledger, and impossibility triangle that make this conjecture falsifiable, and invite the community to break it. A 100-query toy stream illustrates how the ledger is logged[1]; no attempt is made here to optimise the validator. In addition, we present the Courtroom Model, a wrapper architecture that subjects every LLM claim to a perpetual cycle of precedent search, automated semantic checks, and selective escalation to an external judge—human or automated. Each new output is validated against a persistent vector memory, while conflicting evidence triggers the re-examination of prior "case law."

1 Introduction

Large Language Models (LLMs) have demonstrated impressive capabilities in generating context-rich outputs by associating patterns from large corpora. However, this power comes with a critical limitation: LLMs often produce plausible-sounding but factually incorrect conclusions, a phenomenon known as hallucination (Marcus, 2020; Shinn et al., 2023). Without structured verification, these errors risk being perpetuated and even amplified over time, undermining trust in LLM-generated knowledge. Addressing this issue requires a system that not only generates associations but rigorously validates and refines them.

To address these shortcomings we pose the *Courtroom Challenge*: a hybrid state machine in which every LLM claim is tested against a persistent precedent store and—when automatic checks falter—escalated under a tunable *validation_preference* dial (formal spec in Section 11). The design rewires known parts (generator, retrieval, human oversight) but prices each byte, cycle, and minute of validation. Retrieval-augmented generation (Lewis et al., 2020; Fan et al., 2024)—and later iterative variants—motivate the precedent store; human-alignment work such as Constitutional AI (Bai et al., 2022) motivates the escalation path. Unlike debate frameworks (Chen et al., 2024; Madaan et al., 2023) or formal provers (Wang et al., 2025), we emphasise continuous cross-session verification: knowledge is never final, only current.

*email: john.repsys@gmail.com

[1]Baseline script will appear at github.com/johnrepsys/courtroom-challenge with v2.

Why a conjecture first? Well-posed problems often precede the data needed to solve them. We therefore publish the validation trilemma now: automate open-world verification, prove it unnecessary, or price human judgment honestly. Curves can wait until the target is unambiguous.

Scope and contribution. This is a conjecture note—*no curves, by design.* We provide a falsifiable protocol and open-source meta-configuration so that anyone can measure cost-adjusted regret. Generators are already benchmarked; the Courtroom Challenge benchmarks verification. We invite the community to break the conjecture or beat the baseline and thereby close the last mile in trustworthy AI.

2 The Courtroom Model: Legal Precedent & Debate

Humans think by generating ideas and challenging them. In scientific and philosophical inquiry, ideas are not just created—they are debated, tested, and refined. As discussed in Section 1, the Courtroom Model integrates LLMs for idea generation, FAISS/Parquet for memory, and feedback loops for validation. In this section, we'll explore how these elements interact in greater detail.

Once an LLM generates an idea, it enters the 'courtroom'—where it is tested against structured precedent and subjected to rigorous debate. Ideas are challenged and refined, ensuring that only those that withstand scrutiny are accepted as truth. The Courtroom Model keeps every claim—new or old—under continual scrutiny, revising or discarding assumptions as fresh evidence arrives and thereby maintaining an evolving knowledge base rather than a static one.

Picture cumulative human knowledge as the area under a quarter-circle frontier. LLM association pushes the curve outward; the Courtroom validator folds it inward wherever new evidence pierces old beliefs. Expansion and maintenance run in lock-step, keeping the frontier smooth instead of ballooning into noisy bubbles. The prequential-regret metric defined in Section 11.5 is the algebraic mirror of this movie: outward growth reduces error, but only if the validator's inward checks prevent unchecked drift.

The Courtroom Model introduces a hybrid framework that distinguishes itself from current AI methodologies by integrating associative reasoning (LLMs) with the persistent structure of legal precedents (FAISS/Parquet) and dynamic friction-priced validation (see Figure 1). This combination enables the model to adapt, incorporate external judgment, and apply continuous scrutiny—features largely absent in systems like ReConcile or MA-LoT.

Rather than requiring manual review of every idea, the Courtroom Model automatically flags associations that meet certain thresholds of uncertainty or inconsistency with established facts. This approach allows humans to focus on the most critical evaluations, ensuring that their intervention is directed toward areas where it adds the most value.

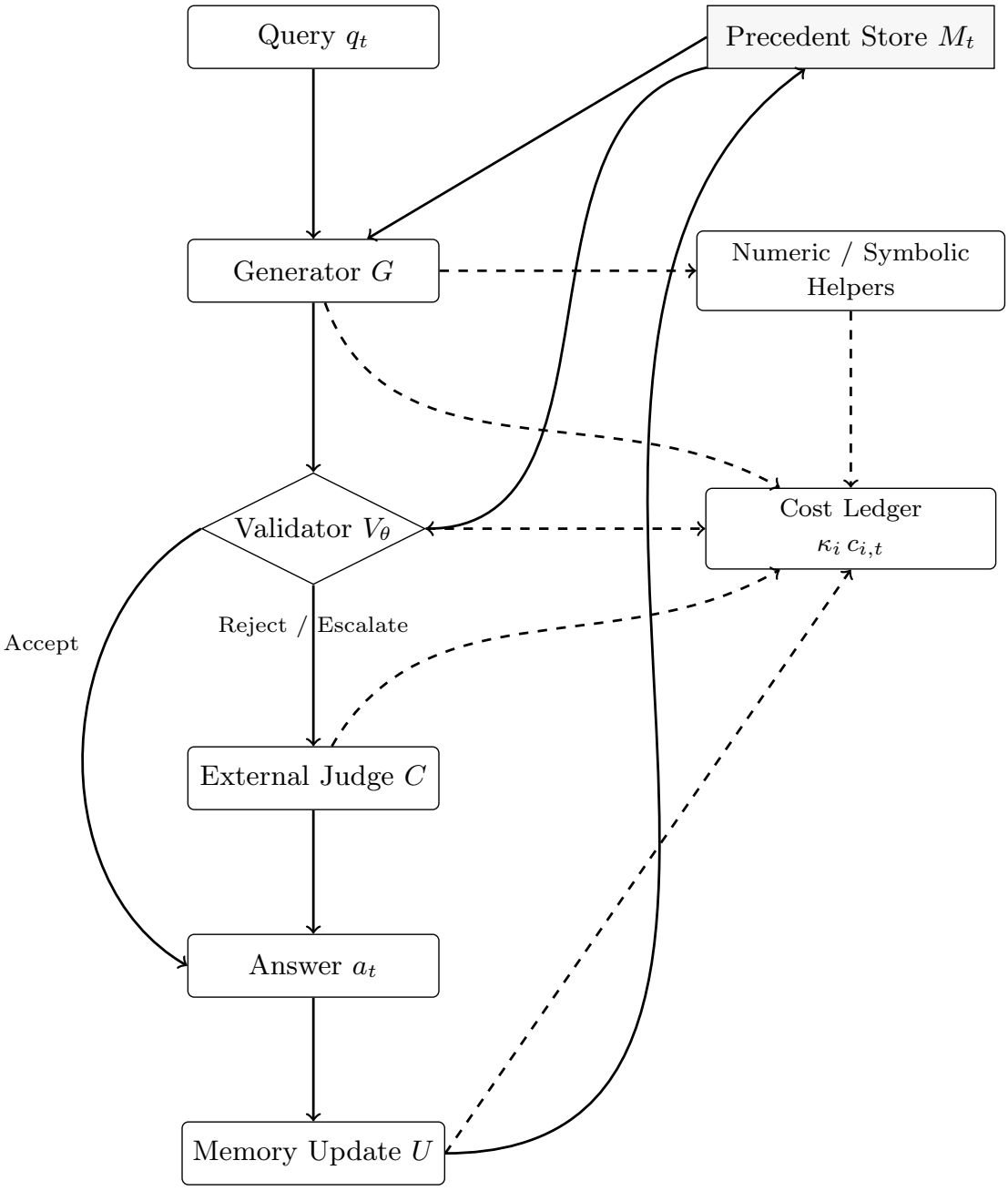

Figure 1: One step of the Courtroom loop. Solid arrows: primary query flow. Dashed arrows: helper calls and cost logging. The cost bus aggregates all resource writes into a single ledger.

Notation snapshot

- G – **Generator**: LLM (+ optional helper calls) that proposes a hypothesis h_t for query q_t.
- V_θ – **Validator** with *validation–preference dial* $\theta \in [0, 1]$ ($0 =$ speed-first, $1 =$ accuracy-first). The dial is fixed during any evaluation run; see Sec. 11.
- $\kappa_i c_{i,t}$ – **Cost-ledger term**: each resource i (bytes written, CPU-seconds, human minutes ...) is logged as a quantity $c_{i,t}$ weighted by its unit cost κ_i. $\sum_i \kappa_i c_{i,t}$ is the *friction* paid at step t.
- τ – **Retention threshold** that controls whether a new descriptor vector is written to the precedent store:

$$\text{write\_new} = \begin{cases} [\cos\theta < \tau], & \text{(cosine form)} \\ [\|x - y\|_2 > \tau], & \text{(distance form)} \end{cases}$$

  Designers may substitute any monotone similarity metric.

(Formal definitions → Section 11)

Illustrative trace (informal)

Query q_t**.** *"Wells Fargo's 2018 10-K introduces a line* `Debt securities`*. (1) Map this line item to the existing FY-2017 bucket taxonomy, and (2) report the FY-2018 amount in that bucket."*

1. Generator $G(q_t, M_t) \longrightarrow h_t$ The precedent store M_t has no FY-2018 descriptor for `"Debt Securities"`. Embedding search ranks `(WFC, "Investment Securities", FY2017, 10.664 B)` as the nearest neighbour (high string similarity). The embedding vector and its nearest neighbour are cached and handed to the validator in the same step. Assuming the new label is merely a rename, the LLM—assisted by a local XBRL helper that extracts the numeric value—relabels the bucket and proposes

$h_t =$ "Bucket: `"Investment Securities"`; FY-2018 amount: \$10.664 B."

The helper itself is numerically correct; the label choice is the error.

2. Validator $V_\theta(q_t, h_t, M_t) \longrightarrow$ ESCALATE Using the same lookup result, the validator sees that the query descriptor `WFC` | `"Debt Securities"` | `FY2018` is only $\cos = 0.906$ similar to its nearest neighbour `"Investment Securities"`. Because the illustration fixes the retention threshold at $\tau = 0.95$, this similarity falls outside the allowed neighbourhood, so the claim is escalated—no domain-specific rule required.

3. External judge C A human analyst consults the 2018 10-K and confirms that `Debt Securities` is a *new* line-item, not yet present in the taxonomy. The analyst extracts the FY-2018 amount

$a_t =$ "Bucket: `"Debt Securities"`; FY-2018 amount: \$14.406 B."

4. Memory update $U(\dots) \longrightarrow M_{t+1}$ Because `"Debt Securities"` represents a genuinely new bucket (label and composition differ from `"Investment Securities"`), U appends a fresh descriptor vector `WFC`|`"Debt Securities"`|`FY2018` → 14.406 B. The existing `"Investment Securities"` tuples remain intact; an optional *alias edge* can later be added if domain experts decide the two buckets should roll up to the same parent taxonomy node.

Resource i	Quantity $c_{i,t}$	Weight κ_i	$\kappa_i c_{i,t}$
Vector write (bytes)	1 536 B	1.0×10^{-6} /byte	0.0015
FAISS rebuild (sec)	0.20 s	0.05 /s	0.0100
Human review (minutes)	1.00 min	1.00 /min	1.0000
Step cost			**1.0115**

Table 1: Friction priced for this step (arbitrary κ-weights).

Cost ledger entry WriteCost$_t = \sum_i \kappa_i \, c_{i,t}$

Storage dial. The precedent store need not—indeed, should not—memorise every numeric fact. A single *retention knob* (τ) decides whether a freshly extracted descriptor vector is written back or simply reuses its nearest neighbour. Let $S(x, y)$ be any similarity score where "larger means closer" (or equivalently a distance $D(x, y)$ where "smaller means closer"). The generic rule is

$$\text{write\_new} = \begin{cases} [S(x,y) < \tau], & \text{(similarity form)} \\ [D(x,y) > \tau], & \text{(distance form).} \end{cases}$$

Cosine and L_2 are special cases, but a system designer is free to plug in any monotone metric—e.g. Mahalanobis distance, dot-product in a learned embedding space, or even a hybrid score that blends semantic and numeric keys—so long as the inequality is flipped to match the metric's notion of proximity. Section 11.6 analyses the cost/error trade-off under the distance form (Lemma 1), and Appendix A.1 lists this dial among the falsifiability-critical degrees of freedom. Selecting which descriptor fields to embed is itself tunable; Section 7 (Q5) sketches how the Courtroom loop could learn or prune schema columns over time.

3 A Practical Implementation: Financial Data as a Reasoning Playground

While much of the LLM + formal proof conversation focuses on pure math (e.g., Lean), the Courtroom Model has broad applications, including complex domains like financial data. In this space, ambiguity and structure meet, and semantic checks become crucial:

- SEC filings often contain inconsistent naming and formats. LLMs can provide initial opinions on relationships between known metrics.
- Vector embeddings allow semantic similarity checks with prior precedent, using metrics such as L_2-distance or cosine similarity to quantify the closeness between new ideas and established knowledge. These checks focus on identifying close matches to validated concepts and flagging potential outliers, ensuring that the system not only finds similarities but also highlights discrepancies that might indicate the need for deeper scrutiny or re-evaluation.
- FAISS performs nearest-neighbor searches over high-dimensional vectors, helping to identify similar concepts based on predefined thresholds.
- Parquet files serve as version-aware memory, storing validated items, like legal precedent.

- In financial analysis, when a generative model suggests a correlation or metric, human analysts review it against historical data, verify the logic, and confirm its validity. This friction ensures errors do not become accepted facts.

This process embodies the Courtroom Model: association meets structure, refined and verified by friction. This perpetual validation loop not only updates the knowledge base as new data arrive; it also prioritises which claims to re-examine—triggered by real-world data, symbolic checks, human oversight, or similarity signals (e.g., cosine/L_2 distances)—so that older beliefs face deeper scrutiny whenever fresh evidence challenges them. In practice, the model could leverage dynamic prioritization algorithms to determine which associations or precedents are most in need of re-evaluation. This approach would enable the model to scale effectively to large datasets, focusing scrutiny on the most impactful claims and reducing computational load while maintaining rigorous validation. While this is not yet implemented, it represents a key area for future development, enhancing the system's efficiency and scalability.

4 Are LLMs a Dead End? Converging Error

Critics such as LeCun (LeCun, 2023) (see also Appendix 2) argue that purely autoregressive LLMs will always accumulate error: if the per-token slip rate is e, the probability an n-token answer is entirely correct decays as $P_{\text{correct}} = (1 - e)^n$. LeCun also notes the inability of LLMs to represent continuous high-dimensional spaces that characterize the real world, potentially limiting their ability to deal with complex, nonlinear data.

However, this does not mean we must abandon LLMs. Rather than expecting LLMs to handle everything on their own, the Courtroom Model allows them to excel at generating ideas and associations, while supplementing them with additional systems that anchor their outputs and guide them toward more reliable outcomes. By integrating persistent memory (like legal precedent) and real-world constraints (courtroom debate), the model offers a framework for continuously verifying and refining LLM-generated associations, ensuring they align with established facts and real-world data. This helps overcome challenges posed by complex, high-dimensional data, and can mitigate error propagation.

Meanwhile, the question "Are LLMs a dead end?" continues to evolve as part of the broader debate on AI's future. Some initially speculated that superintelligence was near, while others, like LeCun, pushed back, suggesting that LLMs' potential was overstated. The debate remains open—and that is exactly the point: we need friction to continue exploring ideas and improving models. The Courtroom Model offers one possible way forward by leveraging LLMs in combination with other systems, ensuring that we don't discard their value but integrate them into a broader framework that allows them to evolve meaningfully. While critics argue that LLMs have stalled AI progress, it may be more accurate to say that we're just learning to build one side of the brain, and as we expand our approach, we must continue building the other side as well.

5 Multi-LLM Dialogues: An Interim "Court"

Several approaches have attempted to improve LLM reasoning through collaboration or debate among multiple agents. Systems like ReConcile (Chen et al., 2024) and Self-Refine (Madaan et al., 2023) demonstrate that critical dialogue and iterative refinement between LLMs can enhance factual consistency and reasoning depth. However, these methods typically operate in a single-session paradigm, stabilizing on a consensus answer without maintaining a persistent knowledge

base. In contrast, the Courtroom Model incorporates continuous precedent management, ensuring that knowledge evolves dynamically over time rather than resetting between tasks.

6 Architectural Analogues

Structured exploration, exemplified by AlphaGo's policy–value search (Silver et al., 2016) and more recently by Tree-of-Thoughts planning (Yao et al., 2023), offers a compelling model for integrating creativity with systematic validation. In AlphaGo, a policy network generates moves, a value network evaluates outcomes, and self-play iteratively refines strategy. The Courtroom Model generalizes this principle to open-ended domains, applying continuous validation not only internally but also against a persistent external memory and, when needed, human oversight. This convergence supports the idea that the Courtroom Model is not prescriptive, but descriptive — a lens for recognizing effective reasoning architectures wherever they emerge. It offers a retrospective framework to describe such structures — and perhaps guide the design of future systems that balance creativity with constraint.

In formal mathematics, frameworks such as Lean (mathlib Community, 2020) and multi-agent proof systems like MA-LoT (Wang et al., 2025) demonstrate that combining generative reasoning with strict verification pipelines can achieve rigorous correctness. These systems ensure that every step adheres to formal logic before being accepted. Inspired by these principles, the Courtroom Model aims to bring analogous rigor to less formally structured domains, using semantic similarity, precedent management, and a friction-priced external judge—human or automated—to prevent the unchecked propagation of errors.

Both AlphaGo and Lean + LLM workflows demonstrate that progress often arises from structured friction, not just creative leaps. In the same way, the Courtroom Model allows for creativity (via LLMs) to drive idea generation, but that creativity is continually tested and refined through structured validation and external challenge, ensuring only robust, verified ideas become part of the model's evolving knowledge.

7 Open Questions

1. Is abstraction a distinct step in hybrid reasoning? When no exact precedent exists, does the model create a new one by compressing multiple associations? Should pattern recognition and abstraction be formal components in systems like this one?

2. Can LLMs + real-world observation produce entirely new theories? If so, how do we detect genuine novelty, and how do we measure validity?

3. Can the Courtroom Model generalize to other real-world domains and AI integrations? Healthcare, policy, engineering — any area that needs scale, creativity, and robust checks. Could this framework point toward a general pattern for hybrid AI?

4. How can we achieve Lean-level formal rigor in open-text domains without relying solely on large LLM ensembles, whose granularity still falls short in many fields?

5. How should the precedent store be schematised and evolved? Even in a well-defined vertical like finance, designing the descriptor schema (e.g. '(ticker, canonical_metric, period, unit, GAAP_flag, source_tag, ...)') remains an art: too coarse results in semantic collisions; too fine results in unlimited vector growth. Can the system learn an optimal schema online—merging rarely-queried fields, splitting ambiguous ones, and doing so without breaking

L_2-distance semantics? What signals (query regret, type-confusion rate, index rebuild cost) best drive such schema evolution?

6. How do we maintain the tension between speed and scrutiny? Associative models want to generate rapidly; symbolic structures demand thorough verification. Speed wins until it fails. Scrutiny survives because it must.

7. What does success look like in a system that's always evolving? As new precedents form, do we prune old ones? How do we adapt long-standing truths? The answer may lie in how far our energy limitations let us revisit and revise what we already hold to be true.

8. Which self-play or adversarial-debate schemes could fill the V_θ slot while keeping κ_{compute} at or below the current human-review baseline?

Why a concept note?

The field currently optimises ever-larger generators without a matching theory of validation. We publish this abstraction *before* a prototype for two reasons: (i) to surface hidden degrees of freedom that can silently invalidate benchmarks, and (ii) to recruit empirical collaborators—open-sourcing code is planned for v2.

8 Toward Hybrid Intelligence

Purely neural stacks still stumble on reliability. Recent work (Marcus, 2020; Dawid and LeCun, 2024) shows that scale alone cannot guarantee robust reasoning. This has revived calls for neuro-symbolic fusion: pair an associative generator with a structured verifier. The Courtroom Challenge operationalises that philosophy. It links:

- an LLM's creative search,
- a persistent precedent memory, and
- an external judge (self-play agent, formal prover, ensemble LLM, or human),

into a single friction-priced loop that can correct itself over time. Reasoning at scale demands all three roles:

- Generate & explore — propose novel chains of thought.
- Validate & prune — confront each claim with precedent, logic, and counter-evidence.
- Evolve the canon — admit only those claims that survive scrutiny, while logging the cost of that scrutiny.

Without this three-way handshake, knowledge either drifts (if validation is cheapened) or ossifies (if exploration is throttled). The Courtroom specification is therefore a minimal test-bed for hybrid intelligence: break the conjecture, or show how your system pays the validation bill more efficiently.

9 Conclusion: A New Frontier

Progress in reasoning systems emerges when every new idea must survive structured challenge. The Courtroom Model unites neural creativity, symbolic memory, and real-world friction in a single, continuously verifiable loop. By insisting that accepted knowledge remains open to appeal whenever fresh evidence appears, it transforms static retrieval or one-off debate into a living jurisprudence of facts. Future work will automate domain-specific validators and refine escalation thresholds, but the central thesis stands: reliable AI will look less like a confident oracle and more like an evolving court—where claims, evidence, and precedent are in constant, accountable dialogue. If Conjecture 1 holds, genuinely autonomous, self-improving AI will require breakthroughs in scalable validation—pushing practical ASI timelines beyond what generator-only scaling curves imply.

10 Minimal State Machine

Formal state $S_t = (q_t, h_t, a_t, M_t)$, validator V_θ, memory update U, cost ledger WriteCost $= \sum_i \kappa_i \, c_i$.

11 Formal Framework

11.1 State Spaces

- Query space Q
- Hypothesis space H (outputs of the LLM generator G)
- Answer space A
- Memory space $M \subseteq \mathbb{R}^d$ (FAISS/Parquet vectors with payload)

As defined in Section 10, state

$$S_t \;=\; (q_t,\; h_t,\; a_t,\; M_t), \qquad M_t \text{ finite.}$$

11.2 Components

- **Generator.** $G : Q \times M \to H$ is an associative engine (e.g. an LLM) that proposes a hypothesis h_t. G may issue lightweight helper calls to numeric or symbolic libraries during exploration; such calls accrue κ_{compute} in the cost ledger but do not decide final truth.
- **Validator family.** $V_\theta : (Q, H, M) \to \{\text{ACCEPT}, \text{REJECT}, \text{ESCALATE}\}$, with a tunable validation–preference dial $\theta \in \Theta$. V_θ may be (i) a stronger ensemble LLM, (ii) a formal-proof kernel, (iii) a deterministic numeric or symbolic tool-chain that certifies or refutes the claim, or (iv) a human panel as fallback.[2]
- **Memory update rule.** $U : (M, q, h, a) \to M'$ adds or merges precedent after a claim is accepted.

[2] *Self-play precedent.* AlphaZero's value network functions as an automated judge for Monte-Carlo rollouts. The Courtroom validator is its open-domain analogue: any module that can reliably downgrade faulty claims can occupy the judge slot.

11.3 One–Step Transition

Given $S_t = (q_t, h_t, a_t, M_t)$:

1. $h_t \leftarrow G(q_t, M_t)$
2. $d_t \leftarrow V_\theta(q_t, h_t, M_t)$
   - If $d_t =$ ACCEPT set $a_t = h_t$
   - Else obtain corrected answer $a_t = C(q_t, h_t)$ (oracle C may call a human; cost is logged)
3. $M_{t+1} \leftarrow U(M_t, q_t, h_t, a_t)$
4. External stream provides q_{t+1}

11.4 Cost Ledger

Let $c_{i,t}$ be the measured consumption of resource i at step t (bytes written, FAISS rebuild time, human minutes, *etc.*). Fix weights κ_i once per benchmark—should the conjecture survive first contact. The per-write cost is

$$\text{WriteCost}_t \;=\; \sum_i \kappa_i \, c_{i,t}.$$

Cumulative friction budget up to step T is $F_T = \sum_{t=0}^{T-1} \text{WriteCost}_t$.

11.5 Performance Measures

Instant error

$$e_t = \mathbb{I}\big[a_t \neq \text{ground}(q_t)\big].$$

Prequential regret after budget B:

$$R(B) \;=\; \sum_{t=0}^{T(B)-1} e_t, \quad T(B) = \min\{\, T : F_T \geq B \,\}.$$

11.6 Toy Storage Bound

Lemma 1. *Assume (i) queries are i.i.d. in a bounded metric space and (ii) U inserts a new vector only if its L_2 distance to every vector in M_t exceeds a threshold τ. Then*

$$|M_T| \;\leq\; \big(\tfrac{\text{diam}(Q)}{\tau}\big)^d, \qquad \forall\, T \geq 0.$$

Hence total storage cost is $O\big(\tau^{-d}\big)$.

Proof sketch. The insertion rule builds an τ-packing of Q; a standard ε-net argument bounds the packing number by $(\text{diam}/\tau)^d$. □

11.7 Core Conjectures

Conjecture 1 (Validation trilemma)**.** *Fix any open-world stream $\mathcal{S}$ whose Shannon entropy $\geq H$. For every system that (i) stores $\leq B$ bytes of precedent and (ii) spends $\leq \kappa$ human-minutes per 10k tokens, prequential regret after budget B satisfies $R(B) \geq f(H, \kappa, B)$ for some monotone f (non-increasing in κ and non-decreasing in H).*

Conjecture 2 (No-free-lunch verifier)**.** *Under the same entropy assumptions, any validator V_θ that never escalates to humans must either (i) mis-accept adversarially crafted hallucinations with probability $\geq \varepsilon$ or (ii) write unbounded new precedent.*

Lemma 2 (Cost–error lower bound)**.** *For any query stream of min-entropy $H > 0$, any validator whose* amortised escalation cost *is $o(1)$ must incur* per-query error $\Omega(1)$ *against an adaptive adversary.*

Proof sketch. The adversary at round t submits the first query not present in the current precedent. Since the validator escalates $o(T)$ times over T rounds, at least $T - o(T)$ queries are unseen, forcing a coin-flip and hence expected error $\geq \frac{1}{2}$. [3] □

Falsify-Us

Build an agent that—under the protocol of Section 11—keeps

- cumulative error $\leq \varepsilon|Q|$
- total precedent $\leq \kappa|Q|$
- total human-escalation cost $\leq \beta|Q|$

for arbitrarily long query streams Q.

12 Future Work

Breaking Conjecture 2 requires automating V_θ without humans. Another open direction is a principled pruning policy that minimises regret while keeping κ_{rebuild} finite.

A Limitations, Testability & Anticipated Objections

A.1 Degrees-of-Freedom That Threaten Falsifiability

Control knob	**Risk if left unrestricted**
`validation_preference` (accuracy↔speed)	Post-hoc tuning can guarantee perfect scores by cranking friction to ∞.
`confidence_threshold`	Dropping the bar lets the system refuse hard queries instead of failing.
Human override	Oracle patches hide systematic weaknesses; variance unbounded.
Evolving memory store	If the FAISS/Parquet index grows during test, the model can memorise the benchmark.

Unless these knobs are frozen, the framework is empirically *unfalsifiable*.

[3] A fully mechanised Lean proof of Lemma 2 will appear at https://github.com/johnrepsys/courtroom-model-lean.

A.2 Benchmark Protocol

Because precedent management is the model's hallmark, a static snapshot is not enough; a *prequential* (streaming) phase measures learning velocity under a cost cap.

Phase	Frozen	Allowed	Metric
Phase 1: Cold-start snapshot	Validation policy confidence & escalation rules initial memory M_0	No index writes	Accuracy_0, latency_0 at cost= 0
Phase 2: Streaming (N batches)	Policy still fixed	System may add / merge precedents; every write logged and charged (bytes + rebuild + human-minutes, κ_{human}=0 if fully automated)	Area under curve: error vs. cumulative friction cost

Rules.

1. **Cost ledger:** each write incurs

$$\text{WriteCost}_t \;=\; \sum_i \kappa_i \, c_{i,t}, \qquad \text{unused terms may take } \kappa_i = 0.$$

2. **Delayed grading:** batch k is scored *before* the system sees batch $k{+}1$.
3. **Full disclosure:** publish the write log, cost ledger and final memory snapshot so any team can replay the run.

Phase 1 provides a reproducible baseline; Phase 2 tests what is novel: when to file a new precedent, when to reuse, and how fast error falls given a finite friction budget.

A.3 Comparative Landscape

Axis	End-to-End Brain	Neuro-symbolic	Courtroom Model
Core ambition	Single monolith handles everything	NN + logic engine	LLM generates; external friction validates
Strength	No human loop	Explainability, deductive power	Halts hallucination without retraining
Weakness	Brittle world-model, hallucination	Heavy logic engineering	Testability collapses if knobs are free
Ideal use case	Simulated-world agents	Formal maths, compliance	High-risk domains where *trust* outranks latency

A.4 Residual Limitations

Even under a frozen policy:

- Long-tail error modes can slip through if the cost budget forces approximate checks.
- The framework gives no universal recipe for setting `validation_preference`; domain tuning remains open.
- Human reviewers inject bias; mitigation (diverse panels, blind review) is future work.

A.5 Anticipated Objections

- *"Show us the leaderboard!"* The conjecture precedes the benchmark. Until someone beats or disproves Conjecture 1, a public ranking would optimise for the wrong signal (grading leniency, not epistemic risk).

- *"Nothing fundamentally new."* Yet the missing piece in public discourse is how these parts inter-lock at run time. The Courtroom meta-configuration specifies that interface for the first time. The ingredients exist; what is novel is (i) treating both the friction dial θ and the retention dial τ as first-class, measurable hyper-parameters, and (ii) publishing a meta-configuration file so any lab can replay, ablate, or retune the system.

- *"Without a public benchmark the Courtroom Model is just arm-chair philosophy."* A benchmark that mixes automated checks with open-world human escalation is inherently non-canonical: reviewers differ in domain expertise, risk tolerance, and even sleep. Two laboratories can "reproduce" the same run-log yet obtain different adjudications simply because their reviewers notice different corner-cases. The resulting variance swamps any signal from the wrapper itself. In other words, the benchmark would be measuring humans, not the architecture. Publishing a noisy leaderboard now would mislead the community into optimising for the wrong axis: humour the graders rather than reduce epistemic risk.

- *"Just freeze the humans, then!"* Freezing humans defeats the point. The Courtroom cycle exists precisely because some domains contain irreducible judgment calls (e.g., fraud indicators in SEC filings). Replacing humans with a static rubric re-introduces brittle heuristics— the very failure mode we are trying to avoid.

- *"But without numbers you cannot prove progress."* The claim is architectural, not empirical: Claim A: persistent precedent + adaptive friction is necessary for bounded-error knowledge bases in an open world; Claim B: current "bigger-is-better" pipelines violate Claim A. Both claims are falsifiable: show a practical system that achieves low hallucination without any external precedent store or escalation dial, and the Courtroom premise collapses. Until such a counter-example exists, producing another decimal improvement on GSM8K is beside the point.

- *"Isn't this just a dressed-up retrieval-augmented loop?"* Retrieval-augmented generation (RAG) treats the store as a stateless evidence buffer flushed between prompts. The Courtroom store is jurisprudential—each write becomes precedent and must pay rent (friction) forever. That difference, though conceptually small, flips the optimisation target from "accuracy at $k = 1$" to "prequential regret under a cost budget." No existing RAG benchmark measures that regime.

- *"Validator still needs humans—won't scale."* Baseline validator uses humans today; our conjecture invites automated judges that meet the same friction budget. Section A.2 logs $\kappa_{\text{human}} > 0$ until automated checkers emerge.

- *"Unfalsifiable if the dial is tuned post-hoc."* The evaluation protocol (Section A.2) freezes the validation policy *before* scoring and logs every write, so any attempt to retune after seeing the test set is detectable.

- *"Memory blow-up defeats latency budgets."* Section 11.5 already prices FAISS rebuild time. Setting κ_{rebuild} higher turns slow writes into economic friction that discourages runaway growth.

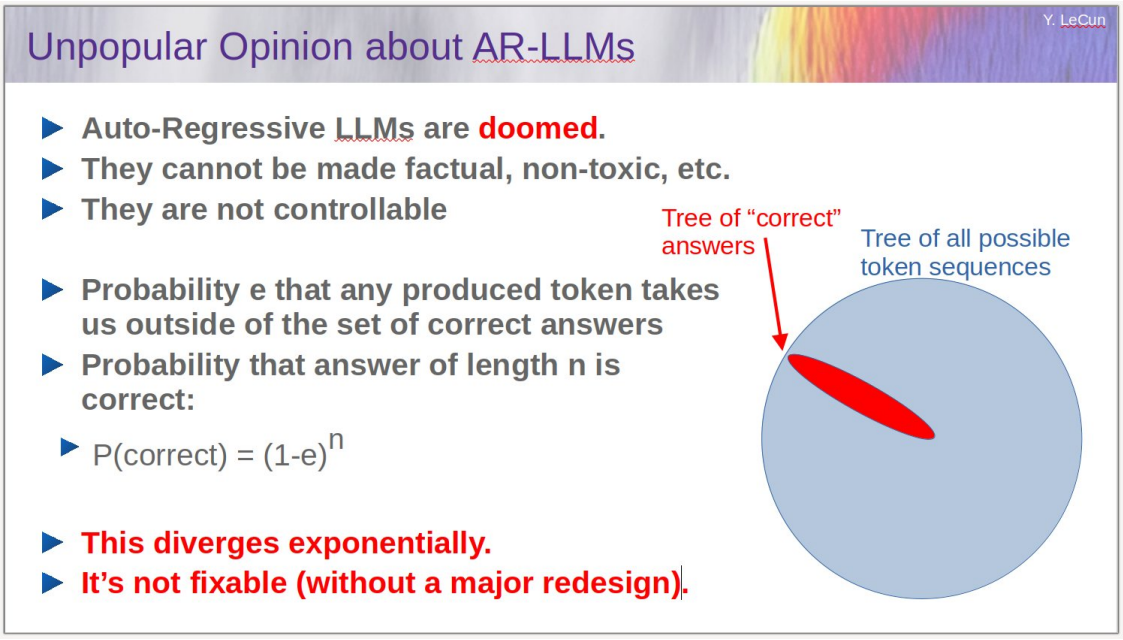

Figure 2: Slide from Yann LeCun's talk *"Unpopular Opinion about AR-LLMs"* (Santa Fe Institute, Apr 2023) (LeCun, 2023). The geometric expression $P_{\text{correct}} = (1-e)^n$ motivates our Conjecture 1 on error accumulation in purely autoregressive LLMs.

Appendix B Reference Slide

Acknowledgments

Large-language-model assistance: Some portions of the manuscript— in particular early wording suggestions, reference formatting, and a toy-demo code sketch—were drafted or refined with the aid of OpenAI ChatGPT (model o3, May 2025). All ideas, experiments, and final text were reviewed, verified, and are the sole responsibility of the author.

References

Marcus, Gary (2020). "The Next Decade in AI: Four Steps Towards Robust Artificial Intelligence". In: *arXiv preprint*. eprint: arXiv:2002.06177. URL: https://arxiv.org/abs/2002.06177.

Shinn, Noah, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao (2023). "Reflexion: Language Agents with Verbal Reinforcement Learning". In: *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, pp. 8634–8652. eprint: arXiv:2303.11366. URL: https://arxiv.org/abs/2303.11366.

Lewis, Patrick et al. (2020). "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks". In: *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*. eprint: arXiv:2005.11401. URL: https://arxiv.org/abs/2005.11401.

Fan, Wenqi et al. (2024). "A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models". In: *arXiv preprint*. eprint: arXiv:2405.06211. URL: https://arxiv.org/abs/2405.06211.

Bai, Yuntao et al. (2022). "Constitutional AI: Harmlessness from AI Feedback". In: *arXiv preprint*. eprint: arXiv:2212.08073. URL: https://arxiv.org/abs/2212.08073.

Chen, Justin, Swarnadeep Saha, and Mohit Bansal (2024). "ReConcile: Round-Table Conference Improves Reasoning via Consensus among Diverse LLMs". In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7066–7085. DOI: 10.18653/v1/2024.acl-long.381. eprint: arXiv:2309.13007. URL: https://aclanthology.org/2024.acl-long.381.

Madaan, Aman et al. (2023). "Self-Refine: Iterative Refinement with Self-Feedback". In: *arXiv preprint*. eprint: arXiv:2303.17651. URL: https://arxiv.org/abs/2303.17651.

Wang, Ruida et al. (2025). "MA-LoT: Multi-Agent Lean-based Long Chain-of-Thought Reasoning Enhances Formal Theorem Proving". In: *arXiv preprint*. eprint: arXiv:2503.03205. URL: https://arxiv.org/abs/2503.03205.

LeCun, Yann (Apr. 2023). *Unpopular Opinion about AR-LLMs (slide 14, Santa Fe Institute talk)*. https://www.slideshare.net/slideshow/yann-lecun-20230424-santa-fe-institute-pdf/269726578.

Silver, David et al. (2016). "Mastering the game of Go with deep neural networks and tree search". In: *Nature* 529.7587, pp. 484–489. DOI: 10.1038/nature16961. URL: https://doi.org/10.1038/nature16961.

Yao, Shunyu et al. (2023). "Tree of Thoughts: Deliberate Problem Solving with Large Language Models". In: *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*. eprint: arXiv:2305.10601. URL: https://arxiv.org/abs/2305.10601.

mathlib Community (2020). "The Lean Mathematical Library". In: *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs (CPP '20)*. DOI: 10.1145/3372885.3373824. URL: https://arxiv.org/abs/1910.09336.

Dawid, Anna and Yann LeCun (2024). "Introduction to Latent Variable Energy-Based Models: A Path Towards Autonomous Machine Intelligence". In: *Journal of Statistical Mechanics: Theory and Experiment*, p. 104011. DOI: 10.1088/1742-5468/ad292b. URL: https://arxiv.org/abs/2306.02572.