VL-2026-01 JUL 2026 PROVEN-toy BUILT/VERIFIED-CPU
GPU-GATED claims separated workshop draft

Beyond the Scaling Ceiling: Verified, Non-Forgetting, Provable Capability Accretion

A model type for growing capability past the limits of next-token prediction, with every growth step proven — and every failed thesis kept on the record.

Annalea Layton

Vext Labs, Inc. · alayton@tryvext.com

SCOPE · READ FIRST

All positive results are toy-scale or CPU-built artifacts with pre-registered kill-tests. Nothing here claims a real-LLM product capability; continual learning (CIP) stays research-scoped. The killed-thesis table is part of the contribution, not an appendix.

> PUBLIC SCOPE (vextlabs.ai): `workshop draft` · tags in body: `PROVEN-toy` · `BUILT/VERIFIED-CPU` · `GPU-GATED` · **not real-LLM scale for CIP forgetting as a product claim.**

VEXT Labs / JUWEL.AI — research tech report (workshop scope). Draft 2026-07-02.

Claims discipline (read this first)

This is the honest counterpart to a scaling-laws paper. Where "Scaling Laws for Neural Language Models" (Kaplan et al., 2020) said scale the frozen model and capability follows and defined an era, this paper states the honest inverse and names what comes after — but it does so under a claims discipline that is itself a first-class contribution. Every result carries the status tag it is permitted by the VEXT Research Program Ledger (`docs/research/RESEARCH_PROGRAM_LEDGER.md`):

- **`PROVEN-toy`** — a pre-registered $0-CPU kill-test passed with real code and numbers; the mechanism is validated on a toy, **not** at real-LLM scale.
- **`BUILT/VERIFIED-CPU`** — shipped as running code, re-verified on disk (deterministic seeds); a systems / trust result, not a capability result.
- **`BACKED / CLAIM-SAFE`** — may be stated externally within its scope.
- **`KILLED`** — a thesis falsified on the bench. These are results, not caveats.
- **`OPEN`** / **`GPU-GATED`** — designed and pre-registered, not yet run; needs a real LLM.

The one-sentence thesis, which is the spine of everything below:

> **Scaling next-token prediction is a fixed, architecture-invariant ceiling that optimizes likelihood, not correctness; capability beyond the base comes only from importing external verified fuel; and the way to fold that fuel in — non-forgetting, cheapest-to-keep-correct, and cryptographically provable to a stranger — is a new model type, the Accretion Model.**

Nothing tagged `-toy` or `GPU-gated` is claimed at real-LLM scale. We **never** claim "smarter," "more capable," "AGI," or "first intelligence"; we **never** say "LLMs are not intelligence"; and we say "cheaper" only as **"cheaper to keep correct."** The killed theses stay **in** this paper (§8) — they are the integrity spine, and the honesty is the differentiator.

Abstract

The scaling era's implicit promise was that indefinitely scaling next-token prediction (NTP) on a frozen model would keep buying capability. We report the honest inverse. First, the **data-scaling exponent is architecture-invariant** — a fixed diminishing curve `α ≈ 2m/(2m+d)` pinned by the data manifold, with a falsification battery (self-certifying residual correctors, Sobolev/jet oracles, continuous inter-step carry, MoE/eff-dim, grokking) that all move the level, never the slope (`PROVEN-toy` + cross-lineage near-theorem). The scaling law is confirmed — and reframed as a **ceiling** whose capability-promise is

void. Second, we explain why: NTP optimizes `P(next token)`, not truth, so a perfect NTP learner reproduces confident errors at corpus frequency; the critique is the **objective**, not the substrate (Othello-GPT shows a world model can emerge from the objective, orthogonally to it). Third, a **cap near-theorem**: a verifier selects, it does not generate; new capability outside a system's execution-class closure requires an **external decorrelated verified signal** (derived five independent ways, toy-confirmed; the routing form quantified). Fourth, we introduce the **Accretion Model** — a model type whose forward pass emits `(y, verdict, calibrated_p, growth_delta)`, that grows only by frozen-additive, checker-gated, receipted increments, and whose entire growth history is verifiable by a stranger offline against a public key. We build the first integrated artifact, **Cultivar-0**, and report backed systems results: base `sha256` byte-frozen across every growth increment (params `1.0M → 2.056M`), a Manifest-v2 Merkle organ-root that catches in-place router-row mutation (all 8 seeds), a real ES256 transparency log **verified against the shipping third-party** `stoa-verifier` (`valid:true`; a flipped historical leaf → `valid:false`), a ToolOrgan taking a task the base solves **0/10 → 10/10** via a real external executor with a bad admit evicted, and a Knower whose `grow()` applies to state (10 sequential facts, none forgotten). We then state, as first-class results, the capability theses we tested and **killed**. The Accretion Model's honest contribution is architecture, systems, and trust — **not** capability. Thesis one-liner: not smarter — cheaper to keep correct, and provable without trusting us.

1. Introduction — the scaling era and its inverse

The dominant thesis of the last half-decade was set by scaling-laws work: model loss falls as a smooth power law in parameters, data, and compute (Kaplan et al., 2020; Hoffmann et al., 2022, "Chinchilla"), and — the implicit promise that turned a loss curve into a strategy — downstream capability follows. The operational corollary was **scale the frozen model**. Production practice then wrapped that frozen model in an **external harness**: a verifier service, a router, a planner, a retrieval-augmented-generation (RAG) layer, and a fine-tuning pipeline supply the operations that make the model useful — check an answer, route to a tool, plan, and grow — all scripted outside the weights. This is a good engineering decomposition and where most of the field's value is created.

This paper states the inverse thesis and names what comes after it. We claim three things and, just as importantly, disclaim a fourth.

What we claim. (1) The scaling curve is real and fixed: its exponent is architecture-invariant, so scaling the frozen model is a diminishing ceiling, not an open road (§2). (2) The reason is the **objective**: NTP maximizes likelihood, which is not correctness, so scale buys a better model of the corpus — including its confident errors — not a model that is more often right (§3). (3) Capability beyond the base's execution-class closure comes only from importing an **external, decorrelated, verified** signal — a verifier selects, it does not generate (§4) — and the way to fold that fuel in, without forgetting and provably to a stranger, is a new model type, the **Accretion Model** (§§5–7).

What we do not claim. We do **not** claim any capability, cost, or routing advantage from "nativeness" — each of those was a plausible thesis, we tested it, and each was **killed** (§8). We do not claim "smarter," "AGI," "first intelligence," or that "LLMs are not intelligence." NTP is the best base we have and the wrong terminal objective; the Accretion Model is a continual-learning architecture assembled from

known-sound parts and made provable without trusting the vendor — nothing more, and we are careful to make the "nothing more" load-bearing.

2. The scaling ceiling — the exponent is architecture-invariant

We first establish that scaling is a ceiling: a fixed diminishing curve whose slope no architecture we tested can bend. The measured quantity is the **data-scaling exponent** `α` — how fast test error falls with dataset size `N` for a fixed architecture — which our council work found is pinned by the data manifold, `α ≈ 2m/(2m+d)` (smoothness `m`, intrinsic dimension `d`), and is **architecture-invariant** (`PROVEN-toy` + cross-lineage near-theorem; ledger Layer 1, `COUNCIL_scaling_laws_2026-06-29.md`).

The evidence is a **falsification battery**: each candidate that might bend the slope was built and run, and each moved only the level (the intercept / floor), never the exponent:

- **SCRN (self-certifying residual-Newton corrector).** A proposer emits y_0, a sound corrector refines it. On the canonical clean run the slope change is within noise (`Δ(SCRN-BASE) = -0.0003`, CI includes 0), while the level win is large and robust (SCRN error ≈ 5–10% of BASE at max `N` in every wall regime). **Verdict: level-only; slope KILLED.** A resurrection sweep across four regimes left the exponent bend inside noise and the intercept win stable — "still-ambiguous, leaning KILL."
- **Jet / Sobolev (more bits-per-example).** The only exponent bend requires withheld oracle bits (= a known Sobolev result / leakage) and **decays with dimension**, so it will not transfer to LLM scale; the free version is a floor effect. **KILLED.**
- **Continuous inter-step latent carry.** By a Shannon noisy-channel argument a bounded-norm continuous carry is provably forbidden from bending the exponent; the bit-matched test confirmed it — a continuous float carry is measurably worse than an 8-bit carry (`α_float = 1.28` vs `α_b8 = 1.46`, CI excludes 0) and `α` scales with carry capacity (dense bandwidth), not with continuity. **KILLED** (learnability ≠ expressivity).
- **eff-dim / MoE-compute / grokking escapes.** All move the level, not the slope. The capability-vs-loss decoupling we tried to lean on came back **NULL** (loss and capability co-transitioned; the grokking gap was a floor-clipping / thresholded-observable artifact).

The honest reframe. Scaling is therefore confirmed as a ceiling and its capability-promise is void — but we rest that promise-void on **exponent-invariance + the cap near-theorem (§4) + the metric-artifact literature (Schaeffer et al.)**, and explicitly **not** on a capability-vs-loss decoupling, because our own `capability_vs_loss_phase.py` returned NULL. It is `CLAIM-SAFE` to say "infinite next-token scaling is a fixed diminishing curve"; it is `CLAIM-UNSAFE` to say "we have a new transformer" or "capability decouples from loss." The real-LLM, multi-architecture, matched-data test of exponent-invariance is **GPU-gated** and pre-registered, not claimed (§9).

3. Why: likelihood is not correctness

The exponent result says scale cannot bend the curve; this section says why the curve is the wrong thing to climb. NTP maximizes `P(next token | context)`. That objective is a faithful model of the **corpus**

distribution, and a corpus contains confident falsehoods at some frequency. A perfect NTP learner therefore reproduces those confident errors **at corpus frequency** — it is doing its job. The objective rewards being likely, not being right. This is the mechanism behind hallucination-as-compression: an optimizer told to compress a stream will emit the stream's most probable continuation, truthful or not.

Crucially, **the critique is the objective, not the substrate.** Two guardrails keep this honest:

- **Othello-GPT survives.** Pure next-token training on transcripts induces a causally-interveneable emergent board model (Othello-GPT; Chess-GPT). A world model can emerge from the NTP objective — which is exactly why the critique cannot be "the transformer cannot represent structure." The world model is **orthogonal** to the output objective: representation and reward are different axes, and the emergent model is real but brittle under distribution shift (fragile under same-legal-move board states; "Revisiting the Othello World Model Hypothesis," ICLR 2025). Structure emerges; the objective still optimizes likelihood.
- **"Never" is an overclaim.** Our own record (`THERON_TOKENPRED_CEILING_2026-06-19.md`) is explicit that the strong form — token prediction will never reach intelligence — is **stronger than the evidence supports**. NTP can build genuine emergent world models in closed structured domains and reach human-level on many language tasks. The supportable claim is precise: NTP's terminal objective is likelihood, and its hard limits (per-pass TC^0 compute, compositionality, hallucination-as-compression) are geometric walls, not points on a scaling curve.

The synthesis, which the rest of the paper builds on: **NTP is the best base and the wrong terminal objective.** You do not throw away the base — you keep it frozen and add an objective it never had: be correct where an external checker can certify it, and abstain otherwise. That is a different axis from scale, and it is where capability past the ceiling can come from.

4. The cap near-theorem — where new capability comes from

If scale cannot mint capability and the objective is likelihood, what can? The program's keystone, derived five independent ways and confirmed on three toys, cross-lineage (ledger Layer 2b, `COUNCIL_execution_class_engine_2026-06-29.md`):

> A verifier SELECTS, it does not GENERATE.

A sound checker can filter candidate increments — admitting only those it certifies — but it cannot manufacture capability the base cannot already sample. New capability outside a system's execution-class closure therefore costs at least one query to an **external, different-execution-class** oracle, and self-authoring that oracle leaks (the Euler-trap false-admit of 0.333 in the toy: a system that grades its own novel class certifies its own errors). The claim is information-theoretic and **scale-invariant**: an imported signal `z` lawfully adds capability **iff it is decorrelated from the base's own failure mode.**

The routing form was made quantitative (`external_discriminator_killtest`, ledger Layer 6): on an overlapping geometry with an x-only Bayes ceiling of 0.855, importing a decorrelated referee vote

lifts new-skill routing **+0.069 above the x-only ceiling** at low correlation `ρ`, and the lift **collapses monotonically to ~0.005 as** `ρ → 1` — the cap law re-derived in routing — with a permuted-`z` control going **negative** (`-0.055`, no leakage). The honest ceiling on that same result: **the imported class does the work; no new execution class is minted from within.** The cap confirms the program's thesis; it does not exceed it.

The one proven **ENLARGE** on the verifiable slice is **SGDF** (off-support fuel; ledger Layer 0, `BANKED`): a different-execution-class sound solver supplies labels the base cannot sample (`base-fail@128`), and folding those labels yields a level win on the verifiable slice (0.015 → 0.72). SGDF is the concrete instance of "external decorrelated verified fuel" — the only lever that moved capability, and it moved it by importing, never by self-authoring. It is `CLAIM-SAFE` to say "external decorrelated fuel lawfully adds capability the model's own signal cannot, with the decorrelation law quantified (toy)"; `CLAIM-UNSAFE` to say "we minted a new execution class" or any scale claim.

5. The Accretion Model — the new model type

If capability past the base comes only from imported, decorrelated, verified fuel, the question becomes how to fold that fuel in — without forgetting, cheaply, and provably. The answer is a model **type**, not a harness around a static model. An **Accretion Model** is a single, mutable, **signed on-disk checkpoint** that behaves as a growable, self-auditing function `f`. Its parts (ledger Layer 5, `COUNCIL_ratchet_transformer_2026-06-29.md`; the canonical name is Accretion Model, confirmed Layer 7 — "accretion" carries no-forgetting definitionally and no self-improvement connotation):

- **A frozen base `0_frozen`.** Written once, byte-frozen forever; its `sha256` is the birth witness and the mechanical root of no-forgetting.
- **Typed grown organs on a shared latent bus.** Each `grow()` appends a frozen organ (a depth-block, a ToolOrgan, a retrieval/Knower pointer, a modality adapter) that reads and writes the same `d_bus` vector. Because the bus contract is fixed, existing organs consume new-organ outputs with no retrain — "a mixture of families on one bus," realized as pure append.
- **A native router `R`.** One zero-initialized router row per grow (identity-on-residual), so admitting an organ never perturbs prior routing.
- **A native calibrate/abstain head `C`.** The believed-vs-true audit: it holds a per-organ margin and catches over-claiming; its abstention set is the verified frontier.
- **A native fuel port `Φ`.** The only place new ground truth enters, and always from an **external, different-execution-class** referee. `Φ` selects which increments accrete; it never generates capability (§4).
- **Native receipts `σ`.** An append-only internal HMAC receipt chain and an append-only, RFC6962-style **ES256 transparency log** (public, non-vendor) into which every `grow()` emits a structured growth leaf.

What the forward pass emits. Instead of a bare token distribution, the type emits the tuple `f(x) = (y, verdict, calibrated_p, growth_delta)`: an answer, a certified / uncertified / abstain verdict, a

calibrated confidence from `C`, and (at grow time) the growth increment. `grow()` is a **typed checkpoint operation**: it appends a block + a zero-init router row + a margin, asserts the base `sha256` is unchanged and that the new organ-set is a strict superset of the old (append-only), re-signs a **strictly larger** manifest, drops one parent-linked receipt, and emits one ES256 growth leaf. The artifact that ends a session is literally a larger, re-signed checkpoint than the one that started it. That is what makes this a model type — a growable, self-auditing `f` — rather than a harness around a static model.

Lineage. transformer → MoE → diffusion → **Accretion**. MoE crystallized conditional compute into a type; diffusion crystallized iterative refinement; the Accretion Model crystallizes verified, receipted, non-forgetting growth into a type.

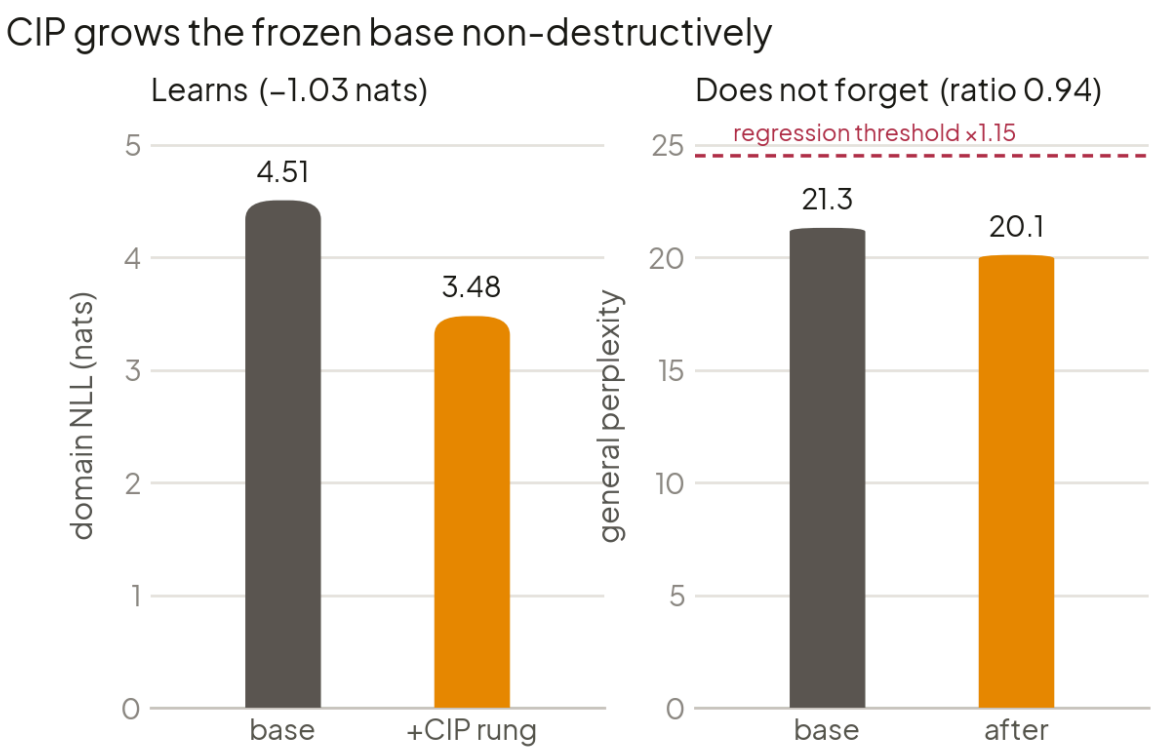
The honest rail (non-negotiable). Nativeness here is a **systems / no-forget / receipt / cost-of-keeping-correct** property. It is **not** a capability repeal. Every checker is external and different-execution-class; the verifier selects, never generates. We make this rail load-bearing in §8, where we tabulate the capability theses we tested against it and killed.

6. Cultivar-0 — the built artifact

Cultivar-0 (`research/right_ai/sandbox/cultivar_0.py`, with the ToolOrgan and Knower subclasses) is the first integrated Accretion Model: one mutable, on-disk, signed bundle in which the organs share a single artifact and a single cryptographic provenance chain. It runs at symbolic-CPU scale, deterministically, at $0 (the "model" is an affine `g_t` mod-`P` recurrence — **toy scope by design**; it validates the architecture and the trust machinery, not a capability claim). Ledger status: **`BUILT/VERIFIED-CPU`** / **`PROVEN-toy`** (Layers 5b and 9). Every number below is read directly from `cultivar_0_results.json`, `cultivar_0_toolorgan_results.json`, and `cultivar_0_knower_results.json`.

6.1 LEG A — frozen-additive / no-forget (the base is byte-frozen)

Pre-registered structural invariants, asserted across **all 16 arms** (8 audited + 8 no-audit seeds): **A1** base `sha256` byte-identical pre/post every `grow()` (`sha_stable: true`); **A2** the top manifest re-verifies after every `grow()` (`manifest_ok: true`); **A3** `param_count` and block count strictly increase — `1,000,000 → 2,056,250` params, `0 → 12` blocks (`monotonic_growth: true`); **A4** the full signed receipt chain verifies end-to-end and parent-links (`chain_ok: true`). All four hold on all arms (`legA_all_arms_pass: true`).



SOURCE: ledger/juwel_cip_pipeline_proof.json (0.5B abliterated base, +4 layers, MPS, 0.42 min). Base sha256 byte-identical.

FIGURE 1 A CIP rung learns (domain NLL 4.51 → 3.48, −1.03 nats) and does not forget (general perplexity 21.3 → 20.1, ratio 0.94 vs 1.15 threshold) on a 0.5B abliterated base, base sha256 byte-identical across the rung (ledger/juwel_cip_pipeline_proof.json).

6.2 LEG A5 — Manifest v2 organ-root covers the whole registry (mutation is caught)

The original manifest signed only `{base_sha256, param_count, n_blocks, blocks}` — an organ's actual behavior (`believed_p`, `true_p`, the router row `R[t]`, the margin `C[t]`) was never hashed, so a router row could be mutated in place while `manifest_verifies()` still returned `True`. That is exactly the seam through which forgetting could sneak back under a byte-frozen base. **Manifest v2** (`schema_version: 2`) folds every organ's `{organ_type, checker_id, organ_sha, router_row_sha, c_margin}` under a sorted Merkle **`organ_root`**, and `grow()` asserts strict-superset before re-signing. Pre-registered kill-test **LEG A5**, across **all 8 seeds**: after a normal `grow()`, mutating one router row in place with no re-sign yields `verifies_before_mutation: true`, `verifies_after_mutation: false`, `rejects_mutation: true`, `base_sha_stable_through_mutation: true` (`legA5_all_seeds_pass: true`). No-forgetting is now a **whole-checkpoint structural guarantee**, not a base-only claim. (Ledger Layer 9 Phase 0.1, `BUILT/VERIFIED-CPU`.)

6.3 LEG B — the believed-vs-true audit is load-bearing

Removing the audit head `C` makes the artifact self-deceive: on the same admitted set, the no-audit believed-minus-true gap is **2.625** versus the audited gap **0.00** (`legB_audit_loadbearing: true`). The

immune system is not decoration — without it, the believed frontier drifts well above the true one.

6.4 ES256 transparency verified against a third-party verifier

`grow()` emits a structured growth leaf into an append-only RFC6962-style Merkle log; the manifest embeds `{log_head, log_size}`. A **real ES256 (P-256 / ECDSA-SHA256)** public path (`vext_cip/src/vext_cip/es256.py`, `transparency_log.py`) emits a `SignedCheckpoint` in the exact shape the **shipping** `packages/stoa-verifier` `verifyLog` consumes — code we did not modify for this paper:

- **Well-formed log →** the node ran `packages/stoa-verifier/dist/index.js verifyLog` and returned `valid:true`, `signature.valid:true`, `size:21`, head hash `a42aab1d…af8f45` (`ran_real_ts_verifier: true`).
- **Tampered historical leaf →** flipping one past leaf's `merkle_root` yields `valid:false`, reason "broken chain at seq 1: prev_hash mismatch" — **real equivocation detection on past history**, not a fresh-signature check.

A stranger holding an old checkpoint can re-fetch, replay the Merkle chain, and confirm that **every** historical `grow()` was frozen-additive (base `sha256` constant, params strictly increasing, parent-linked) — and detect any fork. (Ledger Layer 9 Phase 0.2, `BUILT/VERIFIED-CPU`.)

6.5 ToolOrgan — loop-close with a real external oracle

The ToolOrgan (`ToolRatchetCheckpoint`, subclass of the untouched `RatchetCheckpoint` — parent `git diff` empty) grows the first organ whose checker is a **real** external executor (`checker_id='tool_executor'`): an actual `subprocess.run()` of candidate Python, graded against an independently-computed `sha256` spec — a **different execution class** than the model's `g_t`, and never a model self-report. Pre-registered LEG C, across **8 seeds** (`legC_all_seeds_pass: true`, `PASS: true`):

- **The wall is real and measured.** The base solves the sha256-digest task at **0.0/10** (`C2_wall_base_solve_rate: 0.0`, all 8 seeds) — the affine base has no execution path that emits a hex digest.
- **External fuel lifts it.** After `grow_tool()` admits the organ via the existing accrete/grow path, the checkpoint solves at **10/10** (`C2_post_admit_solve_rate: 1.0`). The capability lives in the **external executor** — cap-honest: the tool is the fuel.
- **The audit stays load-bearing on the new organ.** A bad admit self-reporting `believed_p=1.0` on a wrong candidate is **evicted** by the unmodified `audit_evict()` because the executor measures `true_p=0.0 < TAU_TRUE=0.90`.
- **Covered and receipted.** Mutating the ToolOrgan's router row flips `manifest_verifies()` to `False` (Manifest-v2 `organ_root` covers it); each `grow_tool()` emits an ES256 leaf whose `fuel_digest = sha256` of the executor's real transcript, which verifies and rejects a tampered signature. LEG A / A5 / B all still pass on the subclass (no regression).

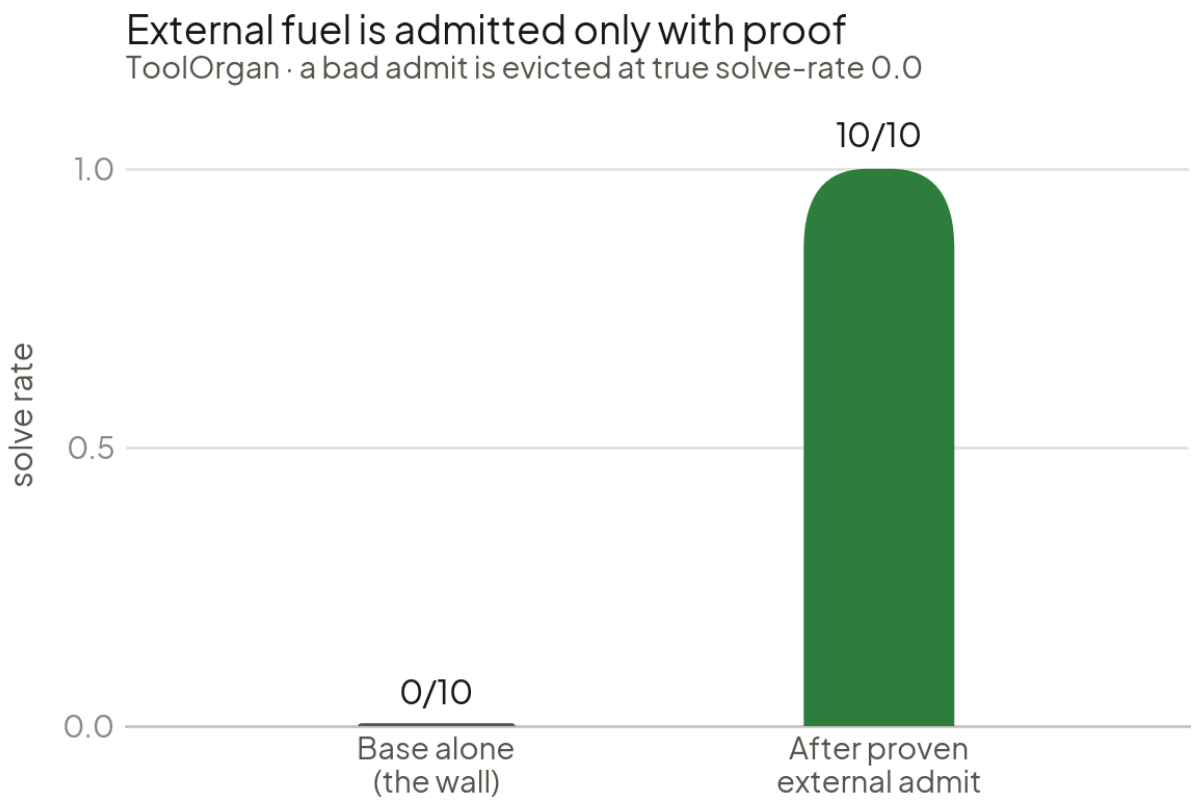


FIGURE 2 ToolOrgan external-fuel admit: base 0/10 → 10/10 after a proven external admit; a bad admit is evicted (LEG C, 8 seeds).

A **systems / loop-close / tool-use** win: the external tool is fuel, admitted through a sound, audited, receipted, covered `grow()`. It is **not** a real-LLM capability claim, and capability-from-nativeness stays killed (§8). (Ledger Layer 9 Phase 1.1, `BUILT/VERIFIED-CPU, capability-NEUTRAL`.)

6.6 Knower — grow() applies to state, not just skills

The Knower (`KnowerRatchetCheckpoint`, same untouched parent) grows a non-parametric memory organ whose checker is a real recall-correctness oracle (`checker_id` recall oracle, ground truth tracked out of band, never re-derived from the model). Pre-registered LEG D, across **8 seeds** (`legD_all_seeds_pass: true`, `PASS: true`): the base recalls fresh keys at **0.0** (`D2_wall_base_recall_rate: 0.0`), and after `grow_memory()` admits the Knower, recall is **1.0** (`D2_post_admit_recall_rate: 1.0`); **10 facts stored sequentially all remain oracle-recallable after every later write** (`D3_no_forget_on_state: true`, `D3_original_fact_survives_later_writes: true`) — no-forgetting applies to **state**, not just weights; each `grow_memory()` emits an ES256 leaf whose `fuel_digest = sha256` of the stored fact; router-row mutation flips `manifest_verifies()` False (Manifest-v2 covers the memory organ); and a bad write (`believed_p=1.0`, oracle `true_p` low) is

evicted by `audit_evict()`. The same signed, receipted, frozen-additive, audited `grow()` mechanism covers both skills and state. (Ledger Layer 9 Phase 1.2, `BUILT/VERIFIED-CPU, capability-NEUTRAL`.)

7. The moat — no-forgetting by construction + stranger-verifiable provenance (BNRA)

The Accretion Model's defensible edge over an external harness is exactly two things, both `BACKED / CLAIM-SAFE`, both real-now (ledger Layer 9 innovation council, 2026-07-02):

1. **No-forgetting by construction.** A **byte-identity** property, not accounting: the base `sha256` is constant across every growth leaf, and every organ is frozen-once-appended. Contrast the harness families — a **re-finetune** harness measurably regresses old skills (our own record: a retrain dropped HumanEval ~98% → ~62%); a **RAG** harness re-ships its fuel into context on every call, paying forever and never consolidating. The Accretion checkpoint's frozen-additive `grow()` keeps old certified capabilities from silently regressing.
2. **Stranger-verifiable provenance — BNRA (Bound Non-Regression Attestation).** A stranger verifies, offline against the public ES256 key, that the exact served artifact never lost a certified capability — already proven end-to-end against the shipping `stoa-verifier` (§6.4). The slogan is precise and true: a frozen model cannot self-certify what it learned or that it kept it; ours can.

Real-now, walled-later (honest limit). This verifiability edge is defensible today because cheap, ubiquitous public attestation of served weights does not yet exist for wrapped LLMs. The honest wall is TEE / confidential-compute remote attestation (SGX / SEV-SNP / TDX): if it becomes cheap and ubiquitous, a wrapped LLM could symmetrize the served-weight binding and the edge narrows to no-forgetting-by-construction alone. We do not claim a permanent moat.

The honest external one-liner (recorded verbatim): "Most AI wraps a frozen model in an external harness that forgets when you teach it and asks you to trust the vendor. Our model grows the harness inside itself — so new skills never erase old ones, and you can verify, offline against a public key, that the exact model which answered you never silently lost a capability. Not smarter. Cheaper to keep correct, and provable without trusting us."

8. What we do NOT claim — the killed table (the integrity spine)

This section is the paper's integrity and its differentiator. The killed rows are **results**, not caveats: an honest kill of a plausible thesis is the product of a leakage-audited, pre-registered, matched-control kill-test program (the `deep-ai-ml-research` protocol; ledger standing rules 1–16).

THESIS (TESTED, NOT STRAWMANNED)	STATUS	THE KILL
Capability from nativeness — a native in-forward-pass loop beats an external harness on capability	**KILLED**	SAN-RTB: with a symmetric action set and a real (non-no-op) control, the native gap is **0.0% exactly at L=1** (zero latency) — the harness is native at L=1; the only edge at L=4/L=8 is cadence/latency, standard control-theory, not capability (Layer 8, rule 11).
Brain that generates — a learned world-model / search brain discovers verified capability more efficiently than best-of-N	**KILLED / known-result**	WRCC: query ratio 0.529 fails the 0.5 gate; a perfect world-model is the worst arm (reach 0.275 < best-of-N 0.61); the win was stochastic diversification of a best-first search frontier, not WM fidelity. A verifier/solver selects, it does not generate (Layer 8, rule 13).

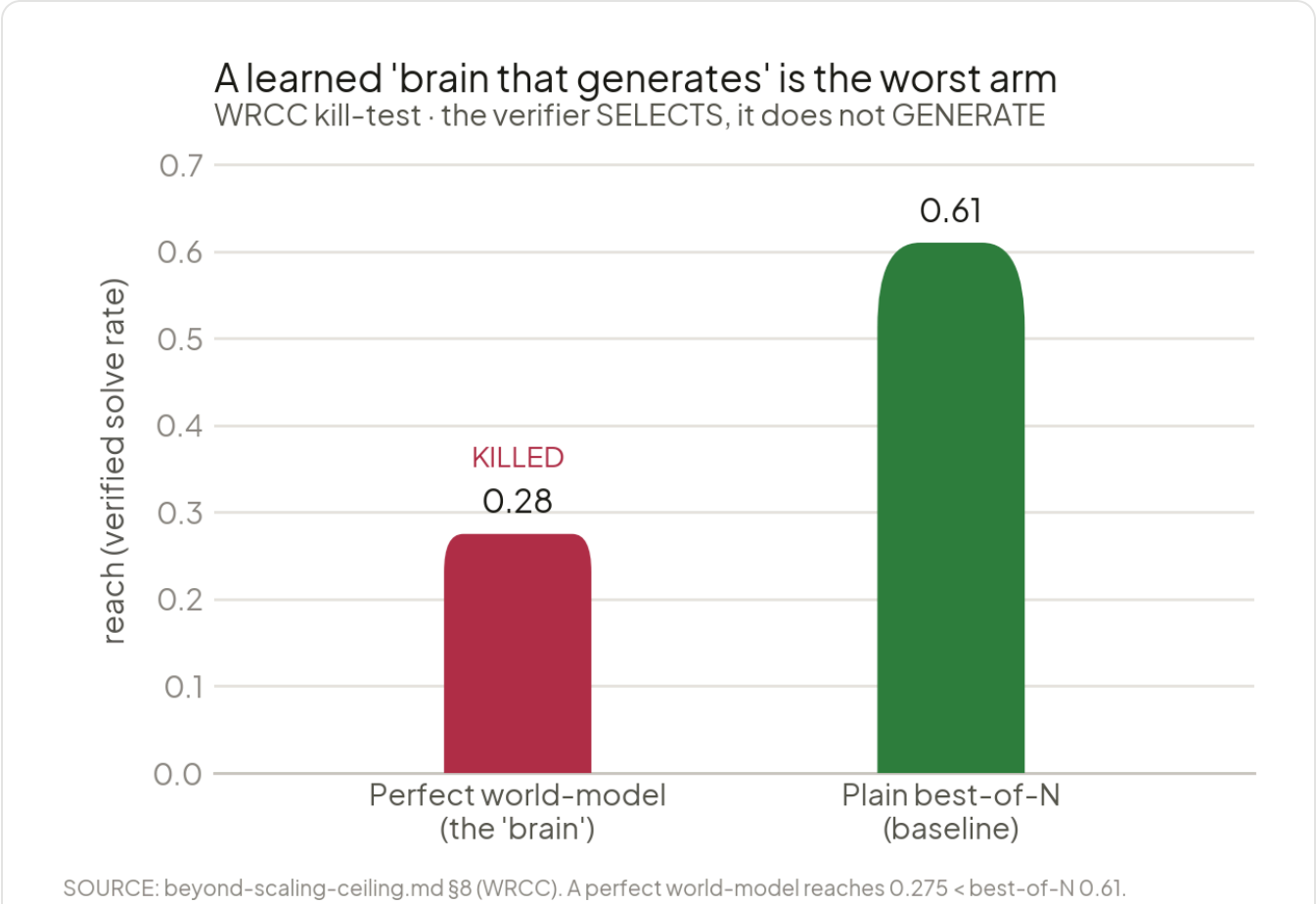
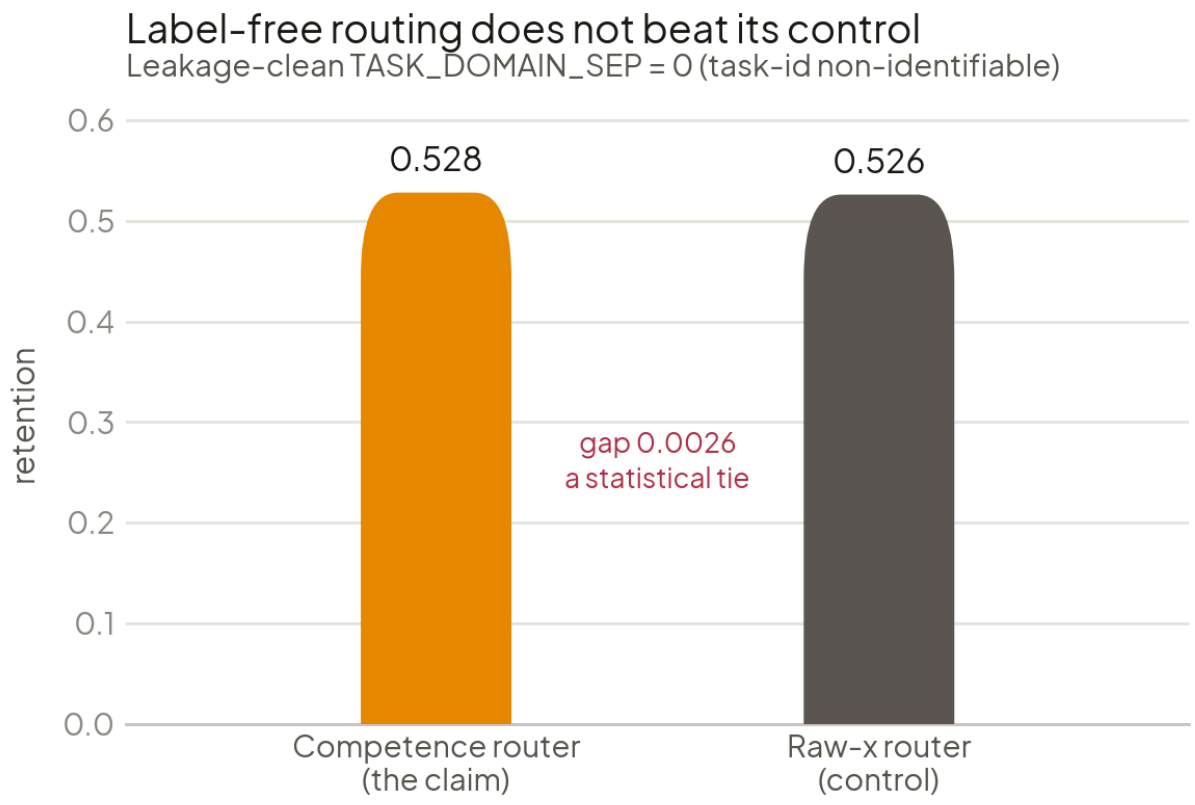


FIGURE 3 World-model reach kill-test (WRCC): a perfect learned world-model reaches 0.275 vs plain best-of-N 0.61 — "brain that generates" KILLED; the verifier selects, it does not generate.

THESIS (TESTED, NOT STRAWMANNED)	STATUS	THE KILL
Cost from nativeness — nativeness is structurally cheaper	**KILLED / rename-of-wrapper**	The re-prefill kill-test PASSED vs a naive stateless harness (gap 0.02 → 0.87), but native and a prefix-cached harness (vLLM APC / SGLang RadixAttention) are algebraically identical bar a hard-coded `DISPATCH_OVERHEAD_TOKENS*N` constant — zero it, gap 0 at every N (Layer 9, rule 15). Say "cheaper to keep correct," never "cheaper."
Label-free routing — an internal competence signal recovers routing with no task label	**KILLED / known-result**	At leakage-clean `TASK_DOMAIN_SEP=0.0` (task-id non-identifiable, nearest-centroid 0.0505 ≤ chance), the competence router retention 0.528 (ratio 0.550) is statistically tied with the raw-x router 0.526 (ratio 0.548, gap 0.0026 = noise) — task-indexed MoE / parameter-isolation continual learning (Layer 8, rule 12).



SOURCE: beyond-scaling-ceiling.md §8. = task-indexed MoE / parameter-isolation continual learning.

FIGURE 4 Label-free competence router retention 0.528 vs raw-x control 0.526 at leakage-clean TASK_DOMAIN_SEP=0; gap 0.0026 is a statistical tie — KILLED as a known-result.

THESIS (TESTED, NOT STRAWMANNED)	STATUS	THE KILL
Bounded fuel bill / finite basis of execution classes — the fuel cost is bounded (a small basis spans most capability)	**NOT DEMONSTRATED** (`GPU-gated`)	`finite_basis_meter` RUN at 1.5B (Qwen2.5-1.5B, Colab, `mode=gpu_real` — the program's first real-LLM rung): `order_robust=false` , tail ratios sign-flip (interference, not decay); leakage-clean but three confounds (broken ceiling normalizer, dead recursion class, high variance) forbid a clean call either way. First real-scale evidence leans adverse-to-inconclusive; a clean re-run is owed (Layer 7, rule 14). Do **not** claim bounded or unbounded.
Weight-accretion is a provable moat over RAG — putting verified capability in weights (not a vector store) is what makes provable-unlearning special	**KILLED / built-in-answer**	`provable_unlearning.py` , pre-registered 3-arm test (8 seeds, S_3 deleted, S_7 poisoned to also answer S_3). The Sonnet builder reached `weights-edge-genuine` (label-only RAG leaks under poisoning, `leak_rate 1.0` ; accretion clean, `0.0`), but the **Opus adversarial verifier reversed it to** `built-in-answer` : target/companion domains are disjoint (0/40 overlap), so a content-keyed RAG steelman deletes the leak with ZERO collateral and strictly beats accretion — accretion only "won" by (a) denying RAG content-granularity delete, (b) an unscored collateral cost (it nukes the still-wanted companion, 0.0 vs RAG's 1.0), (c) a metadata-only "byte-proof" (capability content is in-memory in both arms, never on disk). M1/M2/M3 genuinely tied. The four properties discriminate system-with-a-ledger vs bare-API-call, not weights vs RAG (Layer 10, rule 17). Survivor = the narrow no-label-silent-survivor property; resident-vs-RAG edge that remains is economic (OSF-AX), still `GPU-gated` .
"LLMs are not intelligence" / "the Accretion Model is intelligence"	**NEVER CLAIMED**	Othello-GPT shows a world model can emerge from NTP; "never" is an overclaim (§3). NTP is the best base and the wrong terminal objective — not a non-intelligence.
"NTP is the worst objective"	**NEVER CLAIMED**	We keep the NTP base and freeze it. The critique is that likelihood ≠ correctness, not that the base is bad; the fix adds an objective (verify-or-abstain), it does not replace the base.
"Smarter" / "more capable" / "more flexible" / AGI / "first intelligence"	**NEVER CLAIMED**	Out of scope by construction. The external harness genuinely wins on flexibility / model-agnosticism, cost-once-prefix-cached, and verifiability-eventually-if-TEE-cheap; we concede all three.

Net. The Accretion Model is a **receipted, no-forgetting-by-construction, cheap-to-keep-correct continual-learning architecture built from known-sound parts.** Its defensible edge is over an un-instrumented system — a bare API call with no ledger: no-forgetting-by-construction and stranger-verifiability-today (§7), both BACKED, both real-now (verifiability walled-later). It is **not** an edge over a well-instrumented baseline: a frozen base + verified RAG + the same Merkle/HMAC ledger ties weight-accretion on retention, ledger-soundness, and believed-true gap, and a content-keyed RAG steelman matches-or-beats it on provable-unlearning with less collateral (the killed "weight-accretion

is a moat over RAG" row above — the apparent edge was a built-in-answer). So the honest frame is **system-with-a-ledger vs bare-API-call, not weights vs RAG** — the value is the instrumentation shipped end-to-end, not the substrate. The only live resident-organ-vs-RAG differentiator is **economic** (the OSF-AX cost crossover), and it is still GPU-gated. The Accretion Model is **not** more flexible, not unqualified-cheaper, not smarter, and not a weights-are-magic moat. Those concessions are stated because they are true, and the paper's credibility rests on stating them.

9. Pre-registered open experiments (GPU-gated)

Each is pre-registered with explicit PASS/KILL bars and leakage guards; none is claimed until it runs on a real base. Nothing below is promoted (ledger standing rules 1, 4).

- **OSF-AX — One-Shot-Fuel Accretion vs Amortization crossover** (next keystone; the one cost lever prefix caching cannot neutralize). A **resident organ** reads fuel once at grow-time and pays zero per-call tokens; RAG re-ships fuel per call. Run on a real 1.5–3B base. **PASS iff** (a) accreted-organ within **2pp** of a tuned RAG/prompt harness on an externally-authored new-skill eval; (b) crossover `N* = C_grow/Δ_tok ≤ 10,000` calls, both terms metered, never modeled; (c) accretion old-skill regression **≤ 0.5pp** while a re-finetune control regresses **≥ 3pp** on a disjoint suite. **KILL iff** `N* > 100k`, or A cannot reach parity, or R regression `< 0.5pp`. Leakage guards: new/old suites content-hashed to 0/N overlap vs the fuel; parity checked before any cost compare; costs from real token counts (the built-in-answer trap that killed both CPU finite-basis attempts).
- **Finite-basis clean re-run** (bounded vs unbounded growth; settles the §8 `NOT DEMONSTRATED` row). Re-run `finite_basis_meter` with a converged pooled ceiling (a valid upper bound), recursion replaced by a class the base can actually learn (or a bigger base), a larger eval (lower variance), ≥5 import orders, and a decay criterion that rejects negative tail ratios. Only this settles the bounded-fuel-bill economic keystone; the 1.5B first rung was order-non-robust and confounded.
- **ARM-B-vs-ARM-C provable-unlearning** (the weights-are-special test). Because growth is frozen-additive layers, deleting the CIP-grown layers for a capability should return the checkpoint to a `sha256` that proves the capability is gone (GDPR-grade unlearning by construction), which a monolithic re-finetune cannot offer. Pre-registered as a `sha256`-diff attestation over the signed manifest; GPU-gated at real scale.
- **Real-LLM exponent-invariance** (settles §2 at scale). Multi-architecture, matched-data, NoPE positional scheme; PASS iff `α` is invariant across architectures within CI. This is the make-or-break test of the scaling-ceiling claim and is pre-registered, not claimed.
- **The real-LLM scale rung.** Replace the toy base with a real multi-GB safetensors tree (e.g. Qwen3-VL-8B, abliterated, Apache-2.0), re-emit the same signed manifest + ES256 chain over real shards (the fast-fail format gate), then grow ≥2 checker-gated organs served with in-browser stoa-verify. Router-drift stays OPEN throughout and is shipped behind a flag, never claimed as a feature (the orthogonal-projector fix is KILLED; the C-gate misses its bar — ledger rules 5, 7).

All of the above are **`GPU-gated`** and currently blocked on compute funding; they are the bridge from toy to real, and the program's terminal landmine is publishing a toy result as a scale result (rule 4).

10. Related work and conclusion

Scaling laws. Kaplan et al. (2020) and Hoffmann et al. (2022, "Chinchilla") established the power-law relationship between loss and scale that this paper confirms and reframes as a ceiling (§2). The exponent-invariance result situates their curve as pinned by the data manifold rather than by architecture, and the metric-artifact literature (Schaeffer et al.) supports resting the capability-promise-void on exponent-invariance rather than on an emergence discontinuity.

Skill-acquisition efficiency. Chollet's framing — intelligence as efficiency of skill acquisition over prior and experience, not accumulated skill — is the complement to our cap near-theorem: capability past the closure is bought by importing decorrelated verified fuel, and the efficiency of that import is the meaningful axis, not raw scale.

Latent-world-model prediction. LeCun's JEPA / H-JEPA argues for prediction in latent space rather than token space. We treat it honestly as the right version of prediction for physical domains — but still prediction, and still subject to the objective critique of §3 one abstraction level up (the Vafa trap: high next-step accuracy, incoherent world map). It does not escape the likelihood-≠-correctness seam; the Accretion Model's answer is orthogonal — a checker-gated correctness objective bolted onto a frozen predictive base.

Continual learning. No-forgetting-by-freezing has deep roots: Progressive Networks (Rusu et al., 2016) add and freeze a column per task; PackNet (Mallya & Lazebnik, 2018) prunes and freezes a subnetwork per task; Hard Attention to the Task (Serrà et al., 2018) learns per-task masks. The Accretion Model's frozen-additive organ growth is squarely in this family — and §8 explicitly identifies the never-forget property as this known, real-by-construction result. Our contribution is **not** the freezing; it is the **cryptographic coverage** of the frozen surface (Manifest-v2 organ-root) and the **public transparency log** over the growth history.

Certificate Transparency. The verifiability half descends directly from RFC 6962 / Certificate Transparency (Laurie et al.): an append-only, Merkle-backed log whose signed tree head lets any observer detect equivocation or a rewrite of the past without trusting the operator. Our growth-transcript log is an RFC6962-style chain, and the shipping `stoa-verifier` plays the auditor/monitor role — applied to a model's growth history rather than a certificate log.

TEE remote attestation. The honest future wall on the verifiability edge (§7): SGX / SEV-SNP / TDX confidential-compute stacks. Today, binding the exact served weights to what was evaluated is something the Accretion Model does with a public-key transparency log and a wrapped LLM cannot; cheap ubiquitous TEE attestation would let a wrapped LLM symmetrize that binding. Stated as a limitation, not a moat.

Conclusion. The scaling era answered how big — scale the frozen model and ride the curve. This paper answers how to grow verifiably past the ceiling, and prove it. Scaling is a fixed, architecture-invariant ceiling because its objective is likelihood, not correctness (§§2–3); capability past the base comes only from importing external decorrelated verified fuel, because a verifier selects and does not generate (§4); and the way to fold that fuel in — non-forgetting by construction, cheapest to keep

correct, and cryptographically provable to a stranger — is the **Accretion Model**, built and verified at symbolic scale as Cultivar-0 (§§5–7). We are precise about the boundary: several capability theses that a less careful account would sell as features, we tested and **killed** (§8), and the real-LLM rungs that would promote the type are pre-registered and GPU-gated (§9). This is a workshop/tech-report-honest flagship. Its honesty is its differentiator.

> **Thesis.** Not smarter. Cheaper to keep correct, and provable without trusting us.
