Sound Compounding: A Verifier-Frontier Ratchet for Capability Acquisition and a Believed-versus-True Test for Self-Improvement Leakage

Annalea Layton

Vext Labs, Inc.

alayton@tryvext.com

July 27, 2026

Abstract

Scaling next-token prediction yields diminishing returns: the data-scaling error exponent is empirically and theoretically pinned by the data manifold ($\alpha \approx 2m/(2m+d)$), and we confirm on a falsification battery that no architectural change we tried bends it (companion note). If capability cannot be *scaled* into existence, it must be *acquired and accumulated*. We study the conditions under which a closed verify-grow-reuse loop **compounds** capability — a capability-per-cost curve that does not diminish — rather than plateauing (the documented failure of self-improvement loops). We isolate two mechanisms and one diagnostic. **(1) Discovery via verified search:** the well-known collapse of learned routing over a skill library is avoided by *searching* over a self-grown *certified* library and admitting compositions only under a sound, different-execution-class referee; in a small-scale symbolic setting this reduces next-skill acquisition cost from $O(k^t)$ to $O(k)$ (763× cheaper at depth) with zero false admissions. **(2) Sound slice-growth, and its hard cap:** a system can author new checkers by composing certified ones, growing its verifiable region far beyond its atomic seed (reach 30, zero false admits) — but the growth is **hard-capped at the closure of its seed execution class**: a genuinely new execution class is never self-authorable and requires external off-support fuel. **(3) The believed-vs-true diagnostic:** self-grading loops *self-deceive* (believe they grew far more than they truly did); measuring believed-frontier minus true-frontier against an independent referee exposes this leakage directly (sound: gap 0, false-admits 0; self-grading: gap +1.25, 5 false admits). The naive version of the loop — reusing an *approximate* learned library, or one where the decomposition is supplied externally — is falsified: it either accumulates error down the chain or has the answer built into its setup. The contribution is a precise, honest operationalization of *when* a verify-grow-reuse loop compounds soundly, a reusable leakage test for self-improvement claims, and a boundary theorem-in-spirit (growth = closure of the seed execution class). A real-LLM instantiation is pre-registered.

Keywords: *self-improvement, expert iteration, verifier soundness, compounding capability, leakage audit, pre-registration*

Scope. Every positive result reported in this paper comes from small-scale experiments on symbolic and small-network substrates, run entirely on CPU. They are evidence about a *mechanism* rather than about a working system, and they are not evidence at the scale of a

large language model. The corresponding language-model experiment is pre-registered in Section 7 and has not been run. No claim below should be read as established at scale, and none of them should be cited as settling the question they address.

1 Introduction

Two facts frame the problem. First, scaling is a fixed diminishing curve: under passive sampling from a fixed distribution, the minimax error exponent is set by the data manifold's smoothness and intrinsic dimension, and architecture moves the prefactor, not the slope (Sharma & Kaplan; Bahri et al.; our companion falsification battery kills every architectural escape we tried, including a serial-bit-matched continuous-carry test showing the continuous inter-step channel is dense *bandwidth*, not new *learnability*). Second, reinforcement-learning-from-verifiable-rewards (RLVR) and self-improvement loops largely **re-weight** trajectories the base already samples rather than acquiring new capability (the "sharpen-only" result), and library-learning loops frequently produce *single-use* libraries that do not actually compound.

So the open question is not "what bigger model," but: **under exactly what conditions does a closed loop acquire-and-accumulate capability such that the marginal cost of the next capability does not grow (compounding) — soundly, without forgetting, and without self-deception?** We give toy-scale, pre-registered answers, and a leakage test that any self-improvement claim should have to pass.

2 Background and known traps (what we are *not* re-claiming)

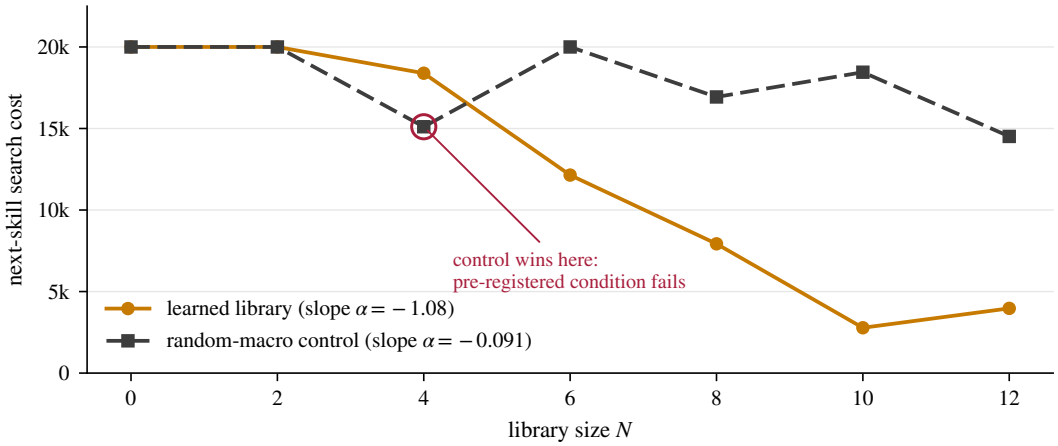

Figure 1. A prior compounding curve that does not survive its own control, reported as a falsification. Next-skill search cost against library size N for a learned wake-sleep library (fitted slope $\alpha = -1.08$) and a matched random-macro control ($\alpha = -0.09$). The learned slope on its own would read as compounding, but the pre-registered condition required the control to be beaten at every N and it is not: at $N = 4$ the random control is the cheaper arm, 15,107 against 18,390. Once the additional-tokens artifact is subtracted the banked verdict is no compounding, so the curve is reported as a negative result rather than as evidence for the mechanism. Symbolic substrate, CPU only, oracle returning a single Boolean per query.

- **Library learning (DreamCoder/Stitch):** MDL-driven abstraction reuse gives a compounding-flavored curve (our prior wake-sleep signal: library prior α=−1.08 vs random −0.09) — but **single-use libraries** are the documented null (e.g. LEGO-

Prover: 1 reuse in 189 proofs). Our discovery result is *in kind* a library-learning result; the contribution is the soundness gate + the bounded-cost separation, not the idea of reuse.

- **Verifier-as-filter (best-of-N + sound verifier):** real but proven — it *selects*, it does not *create* capability the base cannot sample. Our discovery uses verified search, but the *creation* is the per-tier verified authoring; the *compounding* is reuse making search cheap.
- **RLVR sharpens, does not create** (Yue et al.; ReasonMaxxer): the off-support manifold-extension must come from a *different execution class* (SGDF), which RLVR alone cannot supply.
- **CIP / function-preserving growth:** depth-growth with frozen old layers gives no-forgetting capacity, but growth blind to a *verified* signal inflates parameters without earning capability. The novel seam is the *composition*: sound off-support fuel + no-forgetting fold + verified-search discovery + sound slice-growth, plus the believed-vs-true leakage test. (Independent assessment scored the mechanism novelty 5 / soundness 4 / testability 8 — a modest, honest novelty.)

3 Mechanism: the Sound Verifier-Frontier Ratchet

A *slice* is the set of facts a sound referee can certify. The ratchet, per rung: 1. **Fuel.** A different-execution-class sound oracle labels items the current model *fails* at high k (off-support, base-fail@k) — verified-correct trajectories the base cannot sample (SGDF). 2. **Fold.** Train new capacity (a frozen-base + new-LoRA CIP rung) on those labels; old capability is frozen (no forgetting), the new capability is permanent weights. 3. **Discover (next rung).** To reach the next capability, *search* over compositions of the certified library and admit one only if the referee certifies it (verified search, not learned routing). 4. **Grow the slice.** Author a new checker by composing certified checkers; admit only under the independent referee on a fresh adversarial battery. The slice grows. The **soundness invariant**: every admission is gated by a referee that (a) is a different execution class from the learner, (b) never reads test answers, (c) re-derives ground truth independently. The **diagnostic invariant**: track *believed_frontier* (what the system's own slice certifies) vs *true_frontier* (an independent held-out referee). Leakage signature = believed $\gg$ true.

4 Experiments (toy; pre-registered; matched controls; leakage-audited)

All code under `research/right_ai/sandbox/` (+ `verify_comp_*`). Oracles are exact/symbolic (Python permutation algebra / integer predicates), fixed power, never an LLM, never reading test answers; the model runs alone at test. Each result was pre-registered with an explicit pass condition and an explicit falsification condition, and is reported under whichever of the two was met.

4.1 Naive reuse is falsified. Reusing the model's *own approximate* learned modules accumulates error down the chain and still diminishes (own-chain slope -0.088 vs open-loop -0.130). Handing the model the exact decomposition makes the frontier flat (slope $+0.001$) — but a diagnostic shows this is *built-in-the-answer* (frontier == depth by construction), and when the model must *discover* the decomposition by routing, it **collapses to open-loop** (slope $-0.088 \approx -0.066$). Verdict: the result stands but is narrow. "Non-diminishing via reuse" is false unless reuse is sound *and* discovery is solved.

4.2 Discovery via verified search (pass). `compounding_discovery_v2.py` (permutation group $\approx 3.6\times10^6$; 8 seeds). REUSE_SEARCH reaches all tiers at **bounded cost** (per-tier

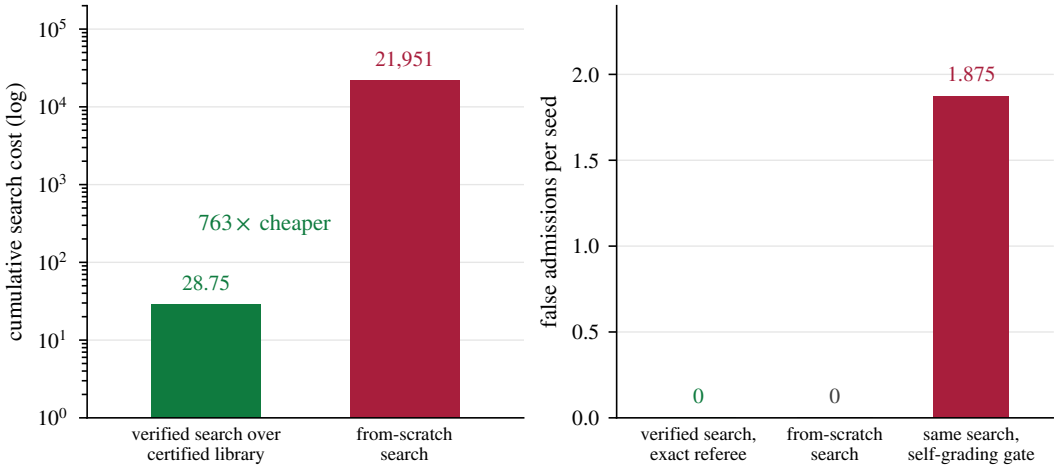

Figure 2. Verified search over a certified library, shown against both of its controls. Left: cumulative cost to reach every tier is 28.75 for verified search over the self-grown certified library against 21,951 for from-scratch search, which stalls at tier five, a ratio of 763. Right: the soundness control, which is the load-bearing one. The exact referee, drawn from a different execution class than the learner, admits zero unsound compositions; replacing it with a self-grading gate over the identical search admits 1.875 per seed. Permutation-algebra substrate of roughly 3.6×10^6 elements, eight seeds, CPU only. The mechanism is library learning combined with verifier-as-filter in kind; what the experiment separates is bounded cost and soundness jointly. No result file was banked for this run, so all four values are transcribed from Section 4.2 of this paper and cross-checked against the experiment log.

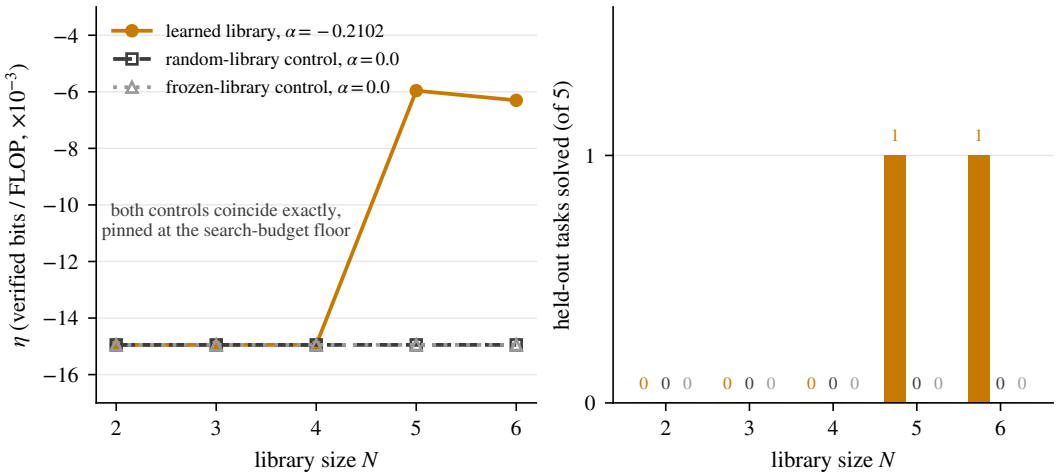

Figure 3. The pre-registered verified-bits-per-FLOP curve, and how weak the positive result actually is. Left: $\eta = \log_2(\text{acceptance})/\text{FLOPs}$ against library size N for the learned library (fitted $\alpha = -0.2102$) and for the two pre-registered controls, a random-library arm in which each additional primitive is an untrained recurrent unit and a frozen-library arm held at $N = 2$ (both $\alpha = 0$). The learned arm improves from -14.9×10^{-3} to -6.3×10^{-3} and exceeds both controls at the largest N, which satisfies the pre-registered condition. Right: the honest reading of that satisfaction. The learned arm solves at most one of five held-out tasks at any N, and both controls solve none at any N, so the controls sit pinned at the 2000-proposal search budget rather than being genuinely outperformed across a range. Held-out tasks are generated at measurement time and the oracle returns only a Boolean. The result file's own stated bound disclaims neural composition from nothing, invention of genuinely new functions, and any transfer to language-model weights.

slope −0.08; cumulative 28.75) with **0 false admits**; from-scratch search **stalls and costs 763× more** (cumulative 21,951); a weak/self-grading gate self-deceives (1.875 false admits). The discovery that collapses under gradient-routing works under verified search over a *certified* library: reuse collapses k^t → k, and exact-referee-gating kills self-grading. *Honest read: library-learning + verifier-as-filter in kind; the contribution is the bounded-cost + soundness separation.*

4.3 Sound slice-growth and its hard cap (pass). `slice_closure_killtest.py` (8 seeds). A seed referee certifies only atomic facts; authored checkers compose it. Result: **reach 30** compound facts beyond the atomic seed with **0 false admits**, but a "rogue" function outside the seed's closure (a new execution class) is admitted soundly **0** times, while a self-grading gate admits **all 20**. **The slice grows soundly to the closure of the seed execution class and is hard-capped there;** a new execution class requires external off-support fuel, not self-authoring.

4.4 Slice-growth ratchet and the believed-versus-true test (pass). `verify_comp_slicegrowth.py`. With a finite slice that must author checkers to extend, gated by an independent different-execution-class referee: **GROW_SOUND** true-frontier 6.75 vs frozen-slice 1.0, **true_slope +0.83 (non-diminishing), 0 false admits, believed==true (calibrated)**; **SELF_GRADE / GROW_UNSOUND** believe 8.0 but truly 6.75 (**gap +1.25, 5 false admits**). The believed-vs-true gap *is* the self-improvement leakage made measurable; the referee rejects a vacuous always-accept predicate, so the lift is not a permissive gate.

5 Limitations and honest scope

1. **Scale.** Symbolic permutation and integer domains, and very small networks. Real LLMs add the deep-feature / position-entanglement obstacle (our prior RoPE blocker) and the open problem of *authoring sound checkers at scale*. No claim transfers to real models until §7 runs.
2. **The cap is real.** The ratchet is uncapped only *within* the seed execution class's closure. Genuinely new capability is **fuel-gated** — it requires an external sound oracle of a different execution class. We do *not* show open-ended capability from nothing; we show sound, no-forgetting accumulation *given a fuel supply*.
3. **Novelty is modest.** The pieces are known (library learning, verifier-as-filter, CIP, SGDF). The contributions are the *composition*, the *soundness gate as the load-bearing element*, and the *believed-vs-true leakage test* — not a new primitive.
4. **Discovery solved only via search.** Gradient-learned routing still collapses (§4.1); our positive discovery result is verified *search*, whose cost we bound on a toy but have not stress-tested at library sizes where search itself could become the bottleneck.

6 The believed-vs-true test as a community tool

We propose `believed_frontier − true_frontier` (system's-own-slice certification minus an independent held-out referee) as a standard audit for any self-improvement / RSI claim: a loop that *believes* it ratcheted while its *true* frontier tracks a sound gate is self-grading, not improving. On this substrate it cleanly separates sound growth (gap 0) from self-grading (gap +1.25). This is cheap and we recommend it as a required control.

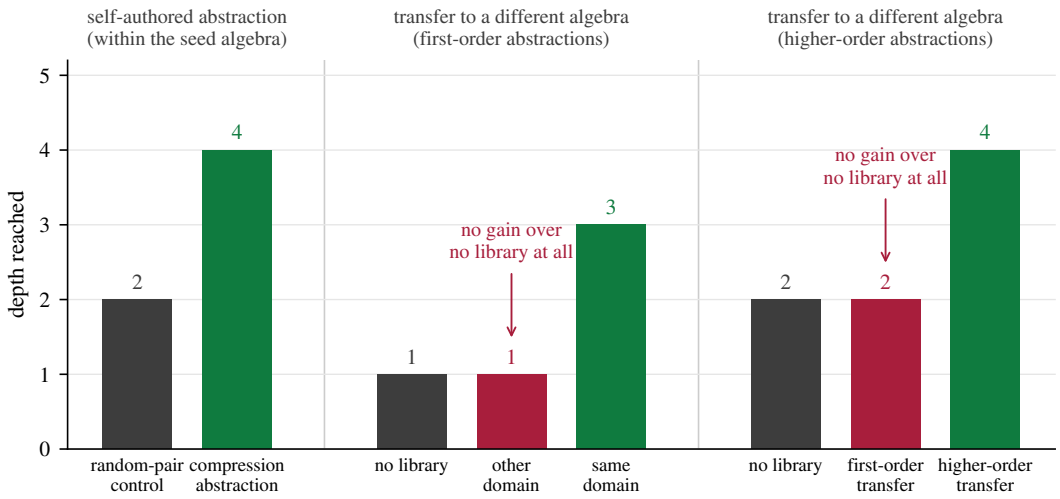

Figure 4. Where reuse buys depth and where it buys nothing, which is what the cap in limitation 2 refers to. Each cluster compares one intervention against its own matched control on the same substrate; the vertical axis is each experiment's own depth measure and is not comparable across clusters. Left: self-authored compression abstractions reach composition depth 4 against 2 for a matched random-pair control, inside the seed algebra. Centre: carrying a first-order library into a different algebra reaches depth 1, exactly what no library at all reaches, while the same-domain library reaches 3. Right: only higher-order abstraction crosses the algebra boundary, reaching a repeat parameter of $k = 4$ against 2 for both no library and a first-order library. The two zero-gain bars are the content of the figure: accumulation is bounded by the closure of the seed execution class and by whatever higher-order structure two domains happen to share, and crossing that boundary requires external fuel rather than further self-authoring. Symbolic substrates, CPU only.

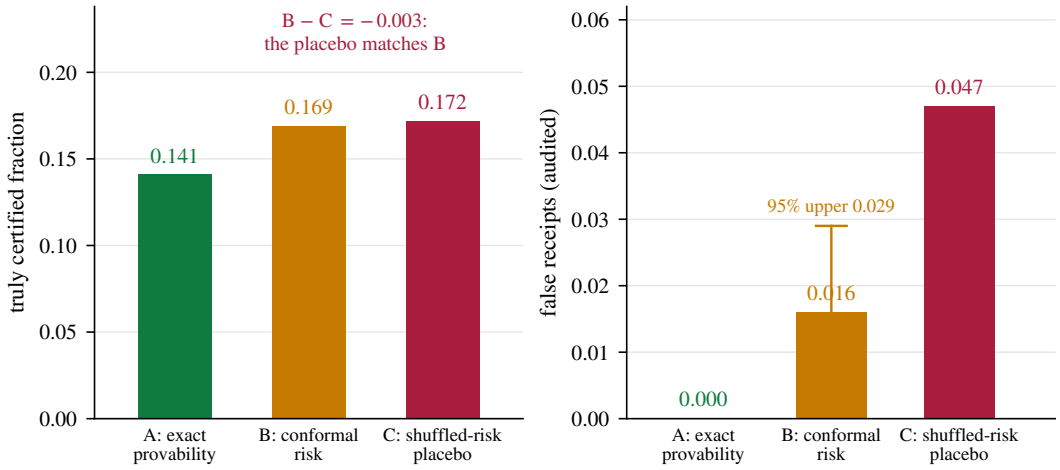

Figure 5. Relaxing the admission gate from exact provability to a calibrated risk estimate fails its own placebo, which is why the ratchet's referee is exact. Left: truly certified coverage is 0.141 under an exact-provability gate, 0.169 under a conformal risk gate, and 0.172 under a placebo in which the risk estimates are shuffled, so the conformal gate's apparent gain of +0.028 over exact provability cannot be attributed to the calibration at all ($B - C = -0.003$). Right: the exact gate produces no false receipts, the conformal gate produces 0.016 with a Wilson 95% upper bound of 0.029, and the placebo produces 0.047. The banked verdict is a falsification. Finite-semantics substrate on CPU, in which actual falsity is computable and is used only to audit receipts after the fact, never to admit a lemma.

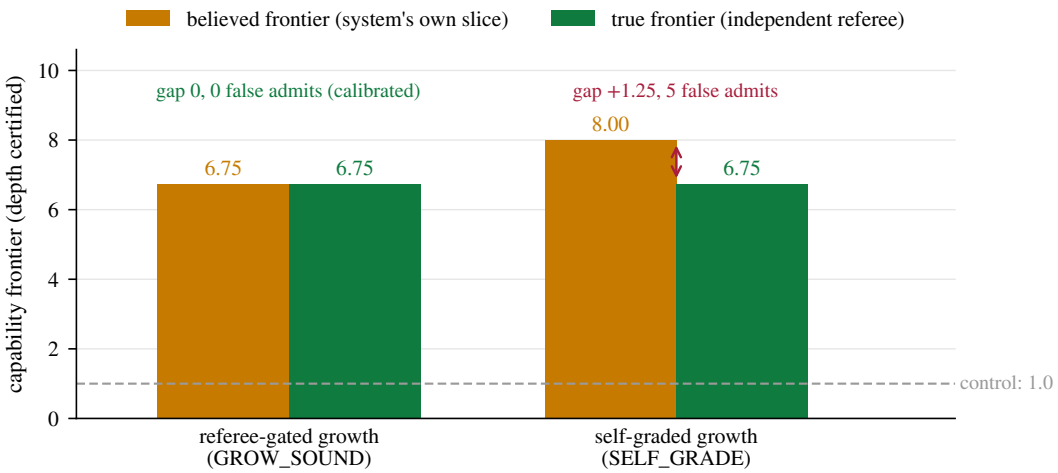

Figure 6. The believed-minus-true frontier gap separates sound growth from self-grading. Under an independent referee of a different execution class the believed and true frontiers agree exactly at 6.75 with zero false admissions. Under self-grading the system reports a believed frontier of 8.00 against a true frontier of 6.75, a gap of +1.25 produced entirely by five false admissions; the admit-on-author variant behaves identically. A frozen-slice control that cannot author new checkers reaches a true frontier of only 1.0, so the growth is measured against a real floor, and the referee was separately audited by confirming that it rejects a vacuous always-accept predicate. Integer-predicate substrate, eight seeds, CPU only; the referee is a supplied sound solver, so the frontier grows to that referee's reach and not from nothing. No result file was banked for this run, so the values are transcribed from Section 4.4 of this paper and cross-checked against the arm table in the experiment log.

7 Pre-registered real-LLM test (the decisive next experiment)

`research/right_ai/colab/real_llm_compounding_rung.py` (free Colab T4). Qwen2.5-0.5B, tiered cumulative modular-arithmetic chains, exact Python oracle. Establish the wall (base fail@8 on deep chains = off-support). **RATCHET** (carry the CIP-LoRA forward, SGDF-fuel each tier) vs **NO_FOLD** (fresh LoRA per tier), matched label budget, with a forgetting probe. **Pass condition:** RATCHET deep-tier pass@1 $\geq$ NO_FOLD + 0.15, forgetting $<$ 0.05, fixed-power Python oracle. **Falsification condition:** reuse does not help, or there is no wall to begin with, or the lift vanishes with answer-only labels (leakage). This is the toy$\rightarrow$real bridge; until it runs, every claim here is scoped to those small symbolic substrates.

8 Conclusion

If scaling is a fixed diminishing curve, intelligence must be *acquired and accumulated*. We give toy-scale, pre-registered evidence that a verify-grow-reuse loop can compound *soundly* — sound discovery at bounded cost, sound slice-growth to the execution-class closure, no forgetting, and a measurable separation from self-grading — while being honest about the cap (new capability is fuel-gated) and the scope (toy until §7 runs). The believed-vs-true test is offered as a reusable guard against the self-deception that has dogged self-improvement research. The mechanism is a modest, honest recombination; its value is that every load-bearing claim has a pre-registered test that it passed and a pre-registered path to real scale.

Reproducibility: all experiments are deterministic, pre-registered, matched-controlled, and leakage-audited under a fixed internal protocol; code and banked result files accompany the paper. Companion: `COUNCIL_scaling_laws_2026-06-29.md` (Layer 1),

`COUNCIL_compounding_2026-06-29.md` (full council log).

9 References

Bahri, Y., Dyer, E., Kaplan, J., Lee, J., & Sharma, U. (2021). Explaining neural scaling laws. *arXiv:2102.06701*.

Bowers, M., Olausson, T. X., Wong, L., Grand, G., Tenenbaum, J. B., Ellis, K., & Solar-Lezama, A. (2023). Top-down synthesis for library learning. *POPL 2023* (*arXiv:2211.16605*).

Ellis, K., Wong, C., Nye, M., Sablé-Meyer, M., Morales, L., Hewitt, L., Cary, L., Solar-Lezama, A., & Tenenbaum, J. B. (2021). DreamCoder: bootstrapping inductive program synthesis with wake-sleep library learning. *PLDI 2021* (*arXiv:2006.08381*).

Sharma, U., & Kaplan, J. (2020). A neural scaling law from the dimension of the data manifold. *arXiv:2004.10802*.

Wang, H., Xin, H., Zheng, C., et al. (2024). LEGO-Prover: neural theorem proving with growing libraries. *ICLR 2024* (*arXiv:2310.00656*).

Yue, Y., Chen, Z., Lu, R., et al. (2025). Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? *arXiv:2504.13837*.