Attention Finds Its Keys

The Findability Frontier: Why One-Shot Block Selection Collapses With Context, and What Training Can — and Cannot — Repair

Mohammad Alsufi* **Connor Scott***

and the Brainsless Research Lab AI Systems Research Group

*Equal lead contribution `research@brainsless.com`

Brainsless Research Lab | Technical Report **BRL-2026-08** | June 2026

Abstract

BRL-2026-07 measured the wall where constant-support decoding stops working: one-shot block-summary selection certifies needle retrieval at 32K and 128K and collapses by 512K, even on a 1M-trained model. This report attacks the wall with three pre-registered hypotheses and refutes all three — and the autopsy mandated by the first refutation yields the law that explains every number. **H1** (hierarchical selection): refuted — two-level bounds-of-bounds selection at constant budget shows *no improvement* over flat — worse at the point estimate (0.17 vs 0.33 recall at 512K, intervals overlapping) — and selection-hit rates show why: the failure is the bound score's ranking, not the width of any decision. **The rank law**: the needle block's score-rank *percentile* is approximately scale-invariant (median 1.6% at 32K, 1.8% at 128K, 3.6% at 512K) while fixed-k survival demands it shrink like $1/n$ — the percentile distribution measured at one length carries to the others (within a length the relation is near-tautological, and we say so), making the 512K collapse arithmetic rather than accident — and the real-prose haystack effect the predecessor measured turns out to be the same law: a single 8.5× rightward percentile shift. **H2** (learned findability, model untouched): a rank-16 query projection trained contrastively on 120 prompts at 128K raises 512K recall 0.27 → 0.47 at k=16 (uncertified at 30 trials/arm — intervals overlap) with the model's forward pass untouched — suggestive signal, refuted regardless against the pre-registered 0.9 bar and 0.6 extrapolation floor. **H3** (adaptive budget from the selector's score gap): refuted — the gap is uncorrelated with need (mean per-window Spearman $\rho \approx 0.08$ across six evaluation windows), so the calibrated policy degenerates into always-spending. Every number is tied to a committed artifact; total compute $56.15 of a $300 cap, itemized. Finding remains the one cost of long context that scales — and it now has a law.

1 The Frontier BRL-2026-07 Left Open

BRL-2026-07 made reading the constant support fast and freed its storage from the GPU, then measured the wall it could not pass: *findability* [Alsufi and Scott, 2026a]. One-shot block-summary selection — score $n/128$ block bounds, keep the top k — certifies needle retrieval at 32K (497/500 at k=8) and 128K (200/200 at k=16) on a long-context-trained model, and collapses by 512K (0.4–0.8 across $k \leq 64$). That fixed-budget sparse attention degrades with context is documented across methods [Nawrot et al., 2025]; what was missing is a mechanism. This report attacks the

wall under the program's rules — pre-registered hypotheses, paired designs, every number tied to an artifact, refutations printed in full [Alsufi and Scott, 2026b] — and finds the mechanism.

Three pre-registered hypotheses (`preregistration/predictions.json`, committed before any hypothesis-bearing measurement — the S0 engine smoke was in flight at commit time, per the ledger): **H1** — the collapse is rank dilution, so hierarchical two-level selection (score $n/2048$ super-block bounds, then only the chosen supers' children; no decision wider than $\sim$512 candidates) restores recall at constant budget. **H2** — findability is *trainable without touching the model*: a low-rank query projection used only by the selector, trained contrastively on synthetic needles, restores recall and extrapolates beyond its training length. **H3** — the selector's top-score gap is a per-step uncertainty signal enabling an adaptive budget.

2 Method

All machinery is BRL-2026-07's engine unchanged (same kernel, cache, prefill, fidelity protocols); selection policies plug in through a single extension point, so every comparison is policy-vs-policy inside one engine. All measurements use `Qwen2.5-7B-Instruct-1M` (the strongest open long-context model at this scale) on single Modal GPUs — H200 for the H1/H2/autopsy suites, H100 for H3 and the wikitext-autopsy pilot — and the *bound score* under study is the predecessor's selector rule, adopted from Quest [Tang et al., 2024]: per block, score $= \mathrm{relu}(q) \cdot k_{\max} + (-\mathrm{relu}(-q)) \cdot k_{\min}$ — an upper bound on any token-level dot product inside the block — evaluated per kv-head and maximized over the GQA group's queries. Readouts per paired trial: end-to-end needle recall (model in the loop) and *selection-hit* (is the needle's block in the keep-set?), which isolates the selector and is the only readout valid at lengths where the dense reference itself fails task retrieval. The hierarchical selector takes bounds-of-bounds over 16-block supers; the learned selector applies $q' = q + B_\ell A_\ell q$ (rank 16, per layer and kv-head, identity-initialized) to the selector's queries only — model logits are bit-identical by construction — trained online: one synthetic-needle prompt, one prefill, one cross-entropy step on the needle block among valid candidates.

3 H1: Hierarchy Does Not Save One-Shot Selection

H1's mechanism story was *rank dilution*: top-k over $n/128$ candidates degrades because the candidate set grows, so keeping every ranking decision narrow — score $n/2048$ super-blocks, take the top 8, then rank only their 128 children — should restore recall at constant budget. The prediction was specific: hierarchical recall ≥ 0.9 at 512K (k=16), refuted below 0.7 or if hierarchy fails to beat flat at equal budget.

	$n = 131{,}072$		$n = 524{,}288$	
policy	recall [95% CI]	sel. hit	recall [95% CI]	sel. hit
dense reference	1.00	—	0.97	—
flat top-16	1.00 [0.89, 1.00]	0.423	0.33 [0.19, 0.51]	0.190
hierarchical	0.87 [0.70, 0.95]	0.368	0.17 [0.07, 0.34]	0.177

Table 1. H1, 30 paired trials per cell (`find_s1_surface.json`). Recall: the engine answers the planted-needle question end-to-end. Selection-hit: fraction of selector calls (28 layers × 8 decode steps, kv-head 0 readout) whose keep-set contains the needle's block. Hierarchy does not restore recall; at 512K its point estimate is below flat's at equal budget (intervals overlap).

H1 is refuted on both registered conditions (recall 0.17 < 0.7; no improvement over flat). The selection-hit column says why, and it is the informative part: flat and hierarchical selection hit the needle's block at numerically indistinguishable rates (0.190 vs. 0.177 at 512K; per-trial hit lists were not retained, so no test is attached — both sit catastrophically below the registered 0.9). The failure is not *where the decision is made* — narrow decisions over at most a few hundred candidates fail exactly as often as one decision over 4,096. The bound *score itself* stops ranking the needle's block highly as n grows. Dilution was the wrong mechanism; hierarchy preserves the score's ordering, so it inherits the score's failure. Per the pre-registration's on-refutation clause, we measured where the needle's block actually ranks.

4 The Rank Autopsy: A Scale-Invariant Percentile

For each trial we capture the selector's queries at the answer step, score every valid block with the same bound rule the selector uses, and record the *rank percentile* of the needle's block — the fraction of blocks that outscore it — per layer and kv-head (`find_autopsy.json`, 12 trials per length).

	$n = 32{,}768$	$n = 131{,}072$	$n = 524{,}288$
median rank percentile	1.59%	1.76%	3.62%
required for top-16 (k/n_b, $n_b = n/128$ blocks)	6.25%	1.56%	0.39%
fraction in top-16	0.64	0.49	0.26
fraction in top-64	0.80	0.60	0.40

Table 2. The autopsy. Per (layer, kv-head): the percentile of blocks outscoring the needle's block, and whether it survives a fixed-k cut. The percentile barely moves while the survival threshold falls 16×. Cuts are quoted over $n/128$ blocks; the percentile denominator is the valid-candidate set ($n/128 - 5$), a $< 2\%$ convention shift at these lengths.

The percentile does not shrink. (Medians; the means run 13.8–19.1% — the distribution has a heavy right tail of layer/kv-heads that rank the needle poorly, and the median is the conservative statistic for the law. In the survival expression below, pct is a fraction.) Across a 16× length range the

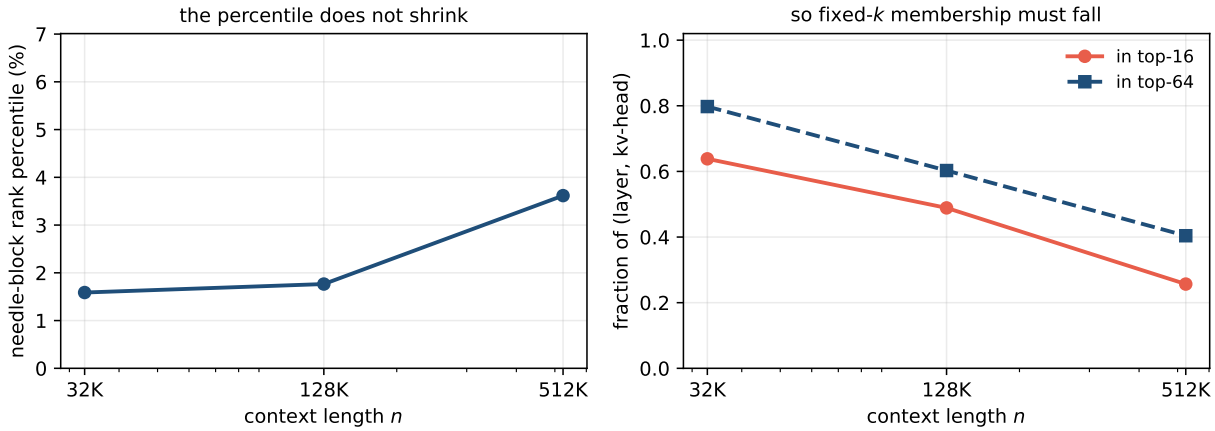

Figure 1. The rank law (`find_autopsy.json`). Left: the needle block's median rank percentile under the bound score barely moves across a 16× length range. Right: because the top-k cut $k/(n/128)$ tightens 16× over the same range, fixed-budget membership falls — the collapse is arithmetic, not noise.

median drifts from 1.6% to 3.6% — approximately scale-invariant, with mild degradation at the long end — while fixed-k survival demands it shrink in proportion to $1/n$: top-16 of 4,096 blocks requires beating the 0.39th percentile. The needle's block reliably outscores ∼96–98% of competitors no matter how many competitors exist, and that single fact generates the entire findability surface:

$$\Pr[\text{block in top-}k] \;=\; \Pr\big[\,\text{pct} < k/(n/128)\,\big]\,,$$

a nearly fixed distribution evaluated at a moving threshold. Two scoping facts a careful reader needs before believing anything downstream. *First*, "in top-k" here is a per-(layer, kv-head) quantity — the fraction of the 112 selector instances (28 layers ×4 kv-heads) whose own top-k contains the needle's block — not end-to-end recall. End-to-end solving needs only *some* retrieval-capable heads to hit, which is why BRL-2026-07 certifies 200/200 solves at 128K while the per-head membership here is 0.49: redundancy across layers converts partial membership into reliable retrieval, until it cannot. That membership→solve link is real, monotone in every measurement we have — and *unmodeled*; the law as stated predicts membership, and its account of recall runs through a link we have not characterized. *Second*, membership and percentiles in each row come from the same trials, so the equation above is close to an identity *within* a length; the law's empirical content is **transport** — the percentile distribution measured at one length predicting membership at others, which is what the table shows across a 16× range, and what §5 exploits.

At 32K the cut (6.3%) sits comfortably above the median — most layer-heads hit (0.64). At 128K the cut (1.6%) lands *on* the median — half hit — and end-to-end recall survives only through the redundancy just described. At 512K the cut (0.39%) is deep in the left tail, the per-layer hit rate falls to ∼0.2 (Table 1), and recall collapses.

The rank law. The needle block's bound-score rank *percentile* is approximately constant in context length (it must shrink like $1/n$ for fixed-k selection to survive; measured, it does not shrink at all). Its absolute rank therefore grows linearly with n, and any fixed-budget one-shot selector — flat or hierarchical — collapses once $k/(n/128)$ falls below the percentile's typical value. The collapse BRL-2026-07 measured at 512K is this law's left tail, not an engineering artifact. *Epistemic status:* the law was found in a pre-registered autopsy but is itself post-hoc; it predicts per-head membership across a 16× length range and is hypothesis-generating for end-to-end recall (the membership→solve link is unmodeled) until it survives a pre-registered transport kill-shot — fit at short lengths, predict long — which is the successor program's first experiment.

The law transports across haystacks, and we tested it. BRL-2026-07 measured that replacing the looped-paragraph haystack with real prose at a *fixed* length (32K) drops recall to 0.58/0.78/0.96 at $k = 8/16/32$ (`s10_filler_wikitext_32768`). Re-running the autopsy on the wikitext haystack (`find_autopsy_wikitext.json`, 12 trials) finds the entire effect in one number: the median percentile shifts from 1.59% to **13.5**% — real prose is an 8.5× more competitive background — with top-16 membership falling 0.64 → 0.38. The shifted distribution accounts for the ladder's shape: the k=32 cut (12.5%) sits just below the wikitext median (13.5%), giving membership ≈ 0.46 (linear interpolation in k between the measured top-16 and top-64 anchors, 0.38 and 0.62) against measured solve 0.96; on the looped haystack a similar membership level (0.49, at 128K/k=16) coincided with solve 1.00 — similar membership, similarly high solve, across two unrelated conditions: the first evidence that the unmodeled membership→solve link is *stable*, on exactly two points, one of them interpolated — and all twelve wikitext trials share one fixed haystack (only needle and position vary), so the percentile draws are not independent across backgrounds. We label this a successful pilot, not a confirmation: 12 trials, interpolated membership at the k=32 cut, and no pre-registration — the registered transport kill-shot remains the successor's E1. In rank-law terms the conclusion stands either way: haystack competitiveness and context growth are the same failure axis — two ways of adding competitors that outscore the needle.

The law also says what a repair must do: nothing that reorders *decisions* (hierarchy, H1) can help, because the percentile is a property of the score. A repair must *compress the percentile* — make the needle's block outscore not 98% but 99.99% of competitors. That is a statement about the query–summary geometry, which is exactly what H2 trains.

5 H2: Findability Is Trainable — But Does Not Extrapolate Far Enough

The rank law says a repair must compress the needle block's score percentile. H2 attempts this without touching the model: a residual low-rank projection $q' = q + B_\ell A_\ell q$ (rank 16, per layer and kv-head, identity-initialized, 458,752 parameters total) is applied to the *selector's* queries only — the model's forward pass never sees it, so the logits computed over any *given* keep-set are bit-identical to the untrained selector's; the projection affects outputs only through *which* blocks get read, which is precisely the quantity being trained (distributional fidelity therefore rides on selection quality and is not separately guaranteed). Training is online and cheap: 120 synthetic-needle prompts at 128K, one prefill each, one cross-entropy step on the needle block among all valid distant candidates per layer/kv-head (`kernel/learned.py`); the contrastive loss falls from 59.7 to 0.52 in minutes of optimization. The pre-registered bar was strict: recall ≥ 0.9 at 512K (4× the training length), refuted if extrapolated recall < 0.6.

The verdict is refuted on the extrapolation clause — with real signal inside. At 512K, k=16, 30 trials per arm on identical prompts (`learn_h2.json`): flat untrained $8/30 = 0.27$ (Wilson 95% $[0.14, 0.44]$; the H1 surface read 0.33 on its own disjoint seed set — each inside the other's interval); flat learned $14/30 = 0.47$ ($[0.30, 0.64]$). The trained projection nearly doubles recall at four times its training length — per-trial pairing was not logged, so we report the conservative unpaired comparison, whose intervals overlap: the gain is consistent and substantial but not certified at this sample size, and either way it sits far below both the 0.9 prediction and the 0.6 refutation floor. The pre-registered eval matrix also included hierarchical-selector arms (tree × untrained/learned); they were lost twice to remote cancellation of the evaluation job and are dropped as a disclosed deviation — §3 independently established that the hierarchical selector is no better than flat at this length, so the flat arms above are the binding test of H2.

Read against the rank law, the result is exactly what partial percentile compression looks like: training at 128K teaches the projection to push the needle block past competitors at the 128K percentile cut (1.6%), but at 512K the cut is 4× tighter (0.39%), and a projection that has never seen 512K-deep score distributions closes only part of that gap. Findability is trainable — the geometry is not frozen — but one cheap contrastive pass at a single length does not buy scale-invariant findability. What would is the open question this program hands to its successor: training across lengths, training the *keys* rather than the queries, or abandoning one-shot scoring altogether.

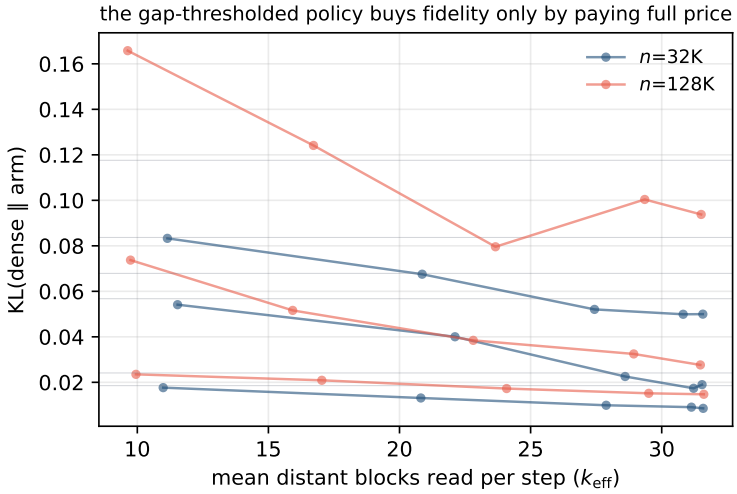

Figure 2. H3's fidelity–budget tradeoff across the τ grid, one curve per evaluation window (`h3_adaptive.json`). The curves slide between the always-8 and always-32 endpoints without a usable knee (one 128K window is mildly non-monotone in the middle): thresholding the gap finds no cheap-but-faithful middle, because the gap does not know where the hard steps are.

6 H3: The Score Gap Is Not an Uncertainty Signal

H3 claimed the selector's own top-score shape is free per-step uncertainty: when the scores between rank 8 and rank 32 sit close to the top of the spread the selector is unsure — spend k=32; otherwise k=8 suffices. Registered predictions: KL within 10% of always-32 at 32–128K with mean per-step $k \leq 14$; refuted if the gap signal is uncorrelated with need.

Protocol (`h3_adaptive.json`): true-text teacher forcing, per-step KL(dense ‖ arm) for always-8, always-32, and the adaptive policy; all arms share one engine and one keep-set buffer shape, so they differ only in budget. The threshold τ was frozen on a calibration window never reused for evaluation, by a rule fixed before any evaluation window ran (smallest mean-k with KL $\leq 1.1\times$ always-32); the rule itself is ours, not in the pre-registration, which specified only the signal and the refutation condition. "Need" is the per-step KL of always-8; the signal is the per-step mean gap.

H3 is refuted, on both registered clauses. The calibration rule could only meet the fidelity bar by giving the budget back: τ^*=0.25 yields mean-$k \approx 31$ of a possible 32 — the "adaptive" policy that satisfies the KL constraint *on the calibration window* is always-32 wearing a costume — and even so, out of calibration the registered "KL within 10% of always-32" clause fails on three of six evaluation windows (1.22×, 1.27×, 1.51×), and its measured throughput matches (79.8 vs. 79.5 tok/s, CUDA-graph-captured decode at 128K — the length is recorded in the launcher, not the artifact's timing block, a logging gap we disclose — vs. 94.5 for always-8 — which also refutes the registered within-5%-of-always-8 cost clause). The reason is the registered refutation condition itself: across six evaluation windows the gap–need rank correlation is $\rho \in [-0.28, 0.27]$, mean per-window ≈ 0.08 — the top of the score distribution carries essentially no information about whether the cheap budget will be sufficient at this step (Figure 2). Spending more when scores look close is spending blind. This closes a tempting design door: per-step budget adaptivity needs a signal from outside the selector's own scores.

7 Limitations

One model (Qwen2.5-7B-Instruct-1M), one needle family (verbatim-code retrieval in a looped paragraph haystack — the easiest haystack for a selector; BRL-2026-07's corpus-variation caveat applies here unchanged), one block size (128). Trials are 12–30 per cell: enough to refute the pre-registered bars decisively, not enough to certify small effects — H2's gain is reported with overlapping unpaired intervals, and per-trial pairing was not logged. The rank autopsy reads one decode step per trial (the answer step) on kv-head 0 readouts for selection-hit; the percentile distribution pools all layer/kv-head pairs. H3's gap signal was evaluated at one $(k_{\text{lo}}, k_{\text{hi}})$ pair $(8, 32)$; richer uncertainty features (entropy over block scores, cross-layer agreement) were not tested and the refutation binds only the registered top-gap signal. All caveats of the shared engine (BRL-2026-07 §Limitations) carry over. **Deviations from the pre-registration, in full:** (i) H1's 1M selection-hit arm was not run — the mechanism it was designed to confirm failed at 512K (recall *and* selection-hit), and the autopsy clause it triggers superseded it; (ii) H1's registered 60 paired trials per cell ran as 30 (cost; the refutation margins dwarf the wider intervals); (iii) H1's scan-cost clause (12× shrink, c_1 within 15%) was mooted by the refutation and never measured; (iv) H2's registered recipe (cached query/summary pairs) was implemented as online single-pass training — same loss, lower memory — and trained at exactly 128K, the bottom edge of the registered 128–256K range (multi-length training is precisely the repair the conclusion recommends); its registered generalization prediction (≥ 0.8 extrapolated) is refuted by the 0.47 above; (v) H2's training-cost clause ($<$ \$25) holds per launch — the ledger's \$27.20 stage entry spans the original run plus two stopped duplicates, \$9–14 each under any accounting — though the stage total overran; (vi) the registered tiering-rig deliverable (assured retrieval at 512K on the 24 GB rig) is dropped — no selector in this report assures 512K retrieval, so there is nothing honest to put on the rig; that deliverable transfers to the successor program. **Pre-registration provenance:** BRL-2026-07 promised registrations as separate commits preceding measurement; this report kept it — `preregistration/predictions.json` entered history in a dedicated commit (together with the runbook; `predictions.json` is untouched since, while the runbook's stage table was corrected post-hoc to describe what actually ran — the amendments are visible in the git history); the smoke-gate artifact entered history 5.5 minutes after it and the first hypothesis-bearing artifact 79 minutes after — but the history is still self-hosted; external timestamping begins with the public release of these repositories.

8 Conclusion

Three pre-registered hypotheses; three refutations; one law. Hierarchy cannot save one-shot selection because the failure lives in the score, not the decision width (H1). The score's own top-end shape knows nothing about when it is wrong (H3). And the score's failure is now quantitative: the needle block's rank percentile is approximately constant in context length when survival demands it shrink like $1/n$ — so every fixed-budget one-shot selector inherits a collapse horizon at $n \approx 128\,k/\text{pct}$. Training the selector's queries moves the percentile (H2's near-doubling at 4× extrapolation) but does not flatten it. The Constant-Support Law still holds — the support *is* constant — but this report sharpens its boundary: reading is constant, storing is tiered (BRL-2026-07), and *finding* is the one cost that still scales, now with a measured law saying exactly how. Collapsing the context window for good means breaking that law — percentile-compressing selection, trained across lengths, or selection that reads more than one shot — and that is BRL-2026-09's problem statement.

Compute statement

All experiments ran on Modal serverless GPUs (H200 for the H1/H2/autopsy suites, H100 for H3 and the wikitext-autopsy pilot; L40S for the smoke gate). Total GPU spend for every number in this report: **$56.15** of a pre-registered $300 cap, itemized per run in `results/cost_ledger.json` — the itemization includes the two cancelled H2 evaluation launches we stopped and count anyway. Verification economics as in the predecessor: any single claim here can be independently re-measured for tens of dollars.

References

Mohammad Alsufi and Connor Scott. Attention pays its bill: From the constant-support law to measured wall-clock. Technical Report BRL-2026-07, Brainsless Research Lab, 2026a. URL https://brainsless.com.

Mohammad Alsufi and Connor Scott. Attention has a type: Constant support, linear finding. Technical Report BRL-2026-06, Brainsless Research Lab, 2026b. URL https://brainsless.com.

Piotr Nawrot, Robert Li, Renjie Huang, Sebastian Ruder, Kelly Marchisio, and Edoardo M. Ponti. The sparse frontier: Sparse attention trade-offs in transformer LLMs. *arXiv preprint arXiv:2504.17768*, 2025.

Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. Quest: Query-aware sparsity for efficient long-context LLM inference. In *International Conference on Machine Learning*, 2024.

A Artifact Manifest

Every number is reproducible from a JSON artifact in the `brl08-results` Modal volume, produced by the named command (run from `papers/brl08`; requires the sibling `papers/brl07` checkout — the engine).

artifact	produced by	used in
`find_s0_32k.json`	`modal run modal_find.py::run_s0`	smoke gate
`find_s1_surface.json`	`::run_s1`	Table 1
`find_autopsy.json`	`::run_autopsy`	Table 2, Fig. 1
`find_autopsy_wikitext.json`	`::run_autopsy_wikitext`	§4 (transport pilot)
`learn_h2.json`	`::run_s2`	§5
`h3_adaptive.json`	`::run_h3`	§6, Fig. 2
`cost_ledger.json`	maintained per run	compute stmt.

Table 3. Number ↔ artifact map. Pre-registered predictions, with refutation clauses, in `preregistration/predictions.json`, committed before any hypothesis-bearing measurement (the S0 smoke was in flight at commit time). The trained projection's `.pt` checkpoint was lost with the cancelled `::run_s2` evaluations and is absent.