Readable but Not Usable

A Preregistered Post-Mortem of a Latent-Workspace Language Model, Including Its Own Novelty Audit

Karthik Yaddanapudi
Independent Researcher
karthikyaddana@gmail.com

Preprint · 5 September 2026

Abstract

We report the complete record of a twenty-day, ten-experiment preregistered attempt to build a latent-workspace language model: a frozen Qwen3-0.6B decoder wrapped in 73M–132M trainable sidecars implementing a root-preserving expert graph, isolated parallel reasoning lanes, and a private diffusion canvas behind a calibrated atomic publication rule. All three mechanisms failed their preregistered gates. Routing was never causally live in any of ten seed-runs; the isolated lanes never passed a distinctness gate; and in the decisive experiment, with the operative premise physically deleted from the decoder's input token ids so the latent channel was the only route from evidence to answer, generation through the channel scored 0.000 on all 99 masked rows on both seeds while a linear probe recovered roughly 0.30 of the withheld premise. The channel was readable but not usable. This constructively extends Lagged Coupling (Xun, 2026), which reports the same read-before-causal dissociation in pretrained models: we built the channel deliberately, made it structurally necessary, repaired every instrument defect our record documents, and the dissociation persisted. The paper includes a novelty audit of its own six architectural claims — every one previously published, with three precisely stated residual cells still unoccupied — and an instrument-failure taxonomy with detectors, covering roughly thirty documented defects in five classes, including gates hardcoded to pass and an expert graph the trainer never called. Three things are explicitly not claimed: the matched plain-LoRA control never ran, so no capability claim is made; the calibrated-abstention thread never beat a free mean-log-probability baseline at its own preregistered bar; and the dense-supervision successor was terminated at 125 optimizer updates, so that question closes undecidable, not falsified.

1. Introduction

Xun's lagged-coupling result reports that the internal representations of language models become linearly readable before they become causally usable by generation, measured across 48 model–checkpoint cells spanning 160M to 12B parameters [1]. This paper is a constructive extension of that finding and of the measurement chain behind it [2, 3, 4]. Where Xun observes the lag in models trained for other purposes, we built a latent reasoning channel on purpose: we wrapped a frozen 0.6B decoder in trainable sidecars, made the channel structurally necessary by physically deleting the premise from the decoder's input token ids on masked rows, supervised it densely from distilled reasoning traces, and repaired every instrument defect we found across ten preregistered experiments. The dissociation persisted. In the decisive measurement, generation through the channel scored 0.000 on all 99 masked rows on both seeds, a linear probe recovered roughly 0.30 of the withheld premise (0.297 and 0.334 by seed, below the preregistered 0.50 decodability bar), and ablating the read-out moved graded accuracy by 0.000 and −0.022. The channel was readable but not usable. The read-before-causal lag documented for pretrained representations therefore also holds for a channel built, supervised, and repaired specifically to be causal. We state this as replication plus extension, never as an independent discovery, and we index it to its regime: it is a claim about a small

sidecar on a frozen 0.6B substrate at a training budget far below published successes, not an architectural law.

The program set out to build one neural model combining three mechanisms: a root-preserving expert graph providing variable-depth conditional expertise; a multi-timescale latent workspace dispatching private briefs to state-isolated lanes; and categorical diffusion refining a private canvas whose only exit was a calibrated, atomic publication decision. Section 2 describes each mechanism as designed and labels what was actually executed, because the gap between the two is itself one of the paper's findings.

Every experiment was preregistered [46]: gate batteries, margins, and binding decision ladders were frozen before each run. Over 20 days (2026-08-17 to 2026-09-05) the program ran ten gated experiments comprising roughly 284 preregistered gate evaluations; about 89 failed, no experiment passed its full battery, and at least eight protocol deviations were disclosed. Most nulls were initially uninterpretable: five classes of instrument defect, about 30 documented individual defects in all, meant that several experiments measured a different system than intended, and instruments were repaired mid-program under fresh preregistrations (Sections 3, 7, and 7.1). The first live channel-capability gate pass in the program's history arrived in Experiment 10's warm phase, on both seeds, in a run that was then terminated on external evidence rather than on its own readings.

That external evidence was a novelty audit the program ran on itself, four days after [1] appeared. Six architectural claims were scored against prior art; all six were already published, with a mean score of 0.63 out of 5 (Section 6). The audit took about 40 minutes and about 200 searches, and would have returned the same verdict before any GPU hour was spent. Three narrow residual cells remain literally unoccupied in the literature; Section 6.1 states them in their narrowest defensible form, with the standing caveat that an unoccupied cell is a contribution only if occupying it produces an effect, which the record shows it did not.

Three claims this paper does not make. First, it makes no capability claim: the matched plain-LoRA control was never run, so ten nulls currently answer "worse than what?" with nothing measured. The authorization for that control drifted over the program's life from unconditional, to postponed, to gated behind a full battery pass that never occurred, to an abort branch — a self-sealing gate that we disclose as a process error, not a protocol virtue (Section 8). Second, it makes no abstention-superiority claim: the calibrated publication machinery was operationally flawless but never cleared its own preregistered significance bar — one pass in six seed-runs, an aggregate advantage of six rows in 300 published at matched coverage — and the design could only detect an effect roughly three times the observed size (Section 5). Third, Experiment 10's dense-supervision question closes undecidable, not falsified: the launched run performed 125 optimizer updates over 2,000 row-presentations, where published successes in this class use 2×10^5 to 1.5×10^7. The mechanism failures (lanes, learned routing) are settled at this scale; the dense latent channel is not, because it was never trained at a competitive budget.

Contributions. This paper offers three things, each stated in its narrowest defensible form.

1. **A constructive extension of lagged coupling [1].** On one frozen 0.6B decoder, two seeds, and 99 masked rows, a purpose-built, densely supervised latent channel under physical premise deletion held probe-recoverable content (0.297/0.334, below its 0.50 bar) while generation through the same channel measured 0.000 — and this survived ten preregistered engineering repairs. The claim is scale- and budget-indexed, and contingent on the unrun control.
2. **A novelty audit of the program's own claims.** Six architectural claims graded against named prior art, all previously published, together with precise statements of the three residual unoccupied cells and an accounting of what the audit cost and when it should have run (Sections 6 and 6.1).
3. **An instrument-failure taxonomy with detectors.** Five defect classes and about 30 documented defects, each with its mechanism, a detector, and the experiment it invalidated (Sections 7 and 7.1) — drawn from one codebase and one team, and offered as a case series with detectors, not a prevalence study.

Reading guide. Section 2 states what was designed and what of it was executed; Section 3 the protocol, budgets, and timeline; Section 4 the ten-experiment record (Sections 4.1–4.3); Section 5 the one thread that kept passing and what it actually shows; Sections 6 and 6.1 the audit and the residual deltas; Sections 7

and 7.1 instrument validity; Sections 8 and 9 limitations and prescriptions; Section 10 concludes. Full gate tables are in Appendix A; the timeline and provenance in Appendix B.

2. What We Set Out to Build

We refer to the system as the workspace model and to the effort as the workspace program. The stated research question was whether a root-preserving, variable-depth expert graph could coordinate isolated parallel reasoning and private block revision while improving verified answer quality at matched active computation. The design was one neural model, not a multi-agent workflow: experts are parameter blocks, lanes are batched tensor paths, recurrence adds executed depth without allocating new blocks, and publication is an exact-snapshot decoding decision. This section gives each mechanism as designed, its one-line intuition, and an execution-status label — *implemented-and-tested* (executed on a launched path, with gate readings on both seeds), *partially-executed* (some component executed; the load-bearing measurement never ran), or *design-only* (no executed path reached it). The labels are established by the code-liveness audit of Section 7.1.

2.1 Root-preserving expert graph

As designed, specialist capacity lives in a fixed typed candidate DAG over low-rank experts — root, then domain, capability, technology, and operation tiers — traversed by a budgeted dispatcher that makes stop-or-expand decisions per functional stage, activating a root-connected subgraph rather than a fixed top-k set [8, 9, 10, 11]. A permanent shared substrate and an always-executed root expert run in every stage, so routing can add specialists but can never remove the generalist [13, 15]; specialists outside the selected subgraph are dormant by default. All conditional computation — graph depth, per-lane recurrent depth, and macrocycle count — was to be charged against one declared budget through an escrow-style ledger that reserves compute for admitted work and a causal fallback before any of it runs. The intuition: a generalist that cannot be routed away, with specialists added only when the dispatcher predicts they will pay for their cost. **Execution status: partially executed.** Flat routed low-rank experts ran with learned routing in the early and middle experiments, and the expert count fell from six to three to two to one across repairs as routing repeatedly collapsed (Section 4); the root-connected graph class existed in the codebase but received zero forward calls on any launched path, and the final experiment's "routing" was a hard-coded gold-label lookup with a detached diagnostic probe (Section 7.1). The escrow ledger was never implemented; it exists only as pseudocode, and we label it design-only.

2.2 Multi-timescale isolated-lane latent workspace

As designed, a dispatcher issues distinct private briefs to state-isolated solver lanes, and every lane runs the same weight-tied transition kernel [31, 32] at three update rates: fast states revise candidate work at every microstep, private slow states consolidate progress at route-window boundaries [26, 27], and a global workspace updates only after verified fan-in at synchronization barriers. Lanes cannot read one another's private states; exchange crosses the barrier only as identified summaries and verification records [28]. In the final instantiation the workspace produced a small set of continuous thoughts — the model's own hidden states recirculated rather than sampled tokens [7, 6] — made visible to the frozen decoder through per-layer, per-head gated key–value projections [30, 5]. On masked rows the premise was physically deleted from the decoder's input token ids, so the workspace channel was the only route from evidence to answer: necessity was a property of the data, not a regularizer. The intuition: incompatible hypotheses survive in parallel because privacy is structural, not learned. **Execution status: partially executed.** The single-lane latent channel is implemented and tested — it is the substrate of the central result of Section 4.3. Multi-lane isolation executed in the early and middle experiments but was never ablated: lanes were retired at a preregistered exit on a lane-summary similarity proxy (six of fourteen readings below the 0.90 bar), so lane diversity was found wanting while lane contribution was never causally measured, and the final experiment ran a single lane.

2.3 Categorical diffusion with calibrated atomic publication

As designed, private canvases are refined by categorical diffusion [33, 34, 35, 36]: a multinomial forward process mixes canvas tokens toward a declared noise distribution, and a denoiser predicts clean tokens conditioned on the committed cache, the global frame, the lane brief, and the private latent state. Sampler stopping is distinguished from semantic commitment, because a stable canvas may be confidently wrong. Commitment freezes an exact token-id snapshot; calibrated commitment, risk, and verification heads [38, 39] decide whether that snapshot is published or a fixed safe fallback is returned, with coverage controlled by a split-conformal threshold [37, 40] rather than a tuned constant, and selective performance scored by a generalized risk-coverage integral [43]. Private canvases never publish directly; publication atomically extends the public cache with the exact buffered ids. The intuition: revision is free until publication, and publication is a single calibrated, auditable decision. **Execution status: split.** The publication gate is implemented and tested: split-conformal thresholding ran throughout the late experiments and was the program's one operationally flawless component, though its calibrated heads never beat the free baseline at their own preregistered bar (Section 5). The diffusion refiner, as a claimed reasoning mechanism, is design-only: the refiner executed inside earlier audit paths, but the preregistered matched-compute control that would isolate its contribution was never implemented in any shipped audit binary, and on the final experiment's executed path the refiner was unreachable dead code. Its marginal contribution was therefore never measured at any scale.

2.4 Substrate and trainable footprint

All ten experiments ran on the same frozen substrate: Qwen3-0.6B-Base, with every backbone weight frozen. Trainable capacity entered only through sidecars — the workspace encoder, the per-layer gated key–value interface, low-rank experts on the decoder's feed-forward projections, and the calibrated publication heads — totalling approximately 73M–132M trainable parameters depending on the experiment. Two facts about this footprint govern how the record should be read. First, a frozen substrate makes every capability comparison contingent on a matched plain-LoRA control, and that control was never run (Section 8). Second, the disclosed trainable counts overstate what was trained: in the final experiment, 57.3% of the disclosed trainable parameters were constructed, registered with the optimizer, and serialized into every checkpoint while receiving zero gradient (Section 7.1). The design of this section should be read with both corrections in hand.

2.5 The case we hoped to make

Honesty about a failed program includes honesty about its ambition, because the ambition explains the program's persistence through ten gate failures. The design was not a small bet. Contemporary frontier systems implement role decomposition *above* the model: the Kimi line holds one mixture-of-experts substrate fixed across K2, K2.5, and K2.6 — reported as 384 experts, 8 active per token, 1 shared — and scales capability through agent swarms that coordinate up to 300 role-specialised sub-agents across 4,000 steps [21, 23, 24, 25]. That division of labour pays real costs: every hand-off serialises context through tokens, no gradient couples the roles, and role composition is fixed by an orchestrator rather than learned per token. The hierarchical bet was that the same division of labour could live *inside* a single forward pass — roles as expert blocks, blended gradedly per token, under a root expert that can never be routed away — so that what a swarm does with prompts and hand-offs, one model would do with parameters and gradients, at what the design intended to be a fraction of the serving cost. The parallel-workspace bet was the same move applied to deliberation: incompatible hypotheses explored in structurally private lanes rather than in separate context windows. The publication bet was that metacognition belongs in the architecture rather than the harness: a model that carries its own calibrated publish-or-abstain decision does not need an external scaffold to know when it is guessing. Had all three paid, the payoff would not have been incremental — single-model deliberation with swarm-like division of labour, conditional compute charged against a declared budget, and answer-level risk control native to the decoder.

The design-level case can be stated precisely, because expressiveness claims about designs are provable even where system claims are not (Table 1). The K-line's mixture-of-experts routes a fixed number of topic experts per token — reported as 8 of 384 with one shared expert through K2.6 and 16 of 896 in K3 [21, 25] — and this design's routing family strictly generalises it on three axes: a variable-depth root-preserving subgraph contains fixed top-k selection as a special case; learned graded role blending contains

the swarm's hard hand-off composition as its one-hot limit; and a breadth-ordered generalist–polymath–specialist ladder contains the flat shared-plus-routed split as its two-tier degenerate form. In-pass role composition also removes costs the swarm design pays by construction: no serialisation of intermediate state through tokens at each hand-off, shared gradients across roles during training, and role mixtures decided per token rather than per orchestration step. These are true statements about design spaces. The record's own verdict on them is equally precise: strictly greater expressiveness bought strictly nothing measurable here, because the added expressiveness is exactly the part that failed to train — a design that strictly generalises on paper and fails its causal-liveness gates in practice is the object this paper documents, and the column that matters, measured quality per unit compute, is empty on our side for want of the control.

Table 1: Design-level comparison with the deployed K-line architecture. The left two columns compare *designs*; the right column records what this program's experiments established about each axis. No row supports a system comparison: the K-line numbers are shipped and measured, ours are a specification whose load-bearing mechanisms failed or went untested at 0.6B scale.

Axis	Kimi K2/K2.5/K2.6/K3 (shipped)	This design (specified)	Measured status here
Role decomposition	Agent-swarm layer above the model; up to 300 sub-agents, 4,000 steps [24]	Inside one forward pass, roles as expert blocks	Learned routing failed causal-liveness gates on every seed-run that tested one (0 of 8 informative readings) at 0.6B
Role composition	Orchestrator-fixed hard hand-offs; context serialised through tokens	Learned graded per-token blend (one-hot as limit case)	Cell unoccupied in the literature; nearest evidence favours hard selection [20]
Active compute	Fixed top-k (reported): 8 of 384 + 1 shared; K3 16 of 896 [21, 25]	Variable-depth root-preserving subgraph; no fixed active count (top-k as special case)	Graph class received zero forward calls on the final launched path
Generalist guarantee	One always-on shared expert [21]	Always-executed root plus breadth-ordered tiers	Untested beyond two experts; reduction traded content for routing
Cross-role learning	None; roles meet only through emitted tokens	Shared backbone, joint gradients across roles	Untested
Deliberation	Parallel sub-agents in separate context windows	Structurally private lanes in one batch	Retired on a similarity proxy; contribution never causally measured
Risk control	Harness-level	In-model calibrated publish/abstain [37]	Operationally flawless; never beat a free baseline at its own bar

What became of each bet, at this scale and budget, is the subject of this paper. Learned routing failed every causal-liveness gate it was given, on every seed-run that tested one (Section 4): settled against the mechanism as built, at this scale. Lane isolation was retired on a similarity proxy without its contribution ever being causally measured: unmeasured, not refuted. The publication heads were operationally flawless and never beat a free baseline at their own preregistered bar (Section 5). The dense latent channel closed undecidable at 125 optimizer updates (Section 4.3). And the one comparison that could have tested any of this against a plain baseline — let alone against a production system — was never run (Section 8). Twenty days on rented dual-T4 sessions against a frozen 0.6B substrate is not the regime in which bets of this size differentiate from their nulls; whether they would at scale is a question this record leaves open. It supports no claim of superiority over any deployed system, and we make none.

3. How the Program Ran

Every GPU experiment was preceded by a written plan that fixed its gate battery before launch: named sub-gates with numeric thresholds, acceptance bands, the two evaluation seeds, and compute budgets, in

the spirit of preregistered NLP research [46]. Each plan also carried a decision ladder, an ordered list of binding rungs pairing a pre-committed failure condition with a prescribed structural response, so that a failed gate forced a decision rather than a rerun. The ladders were consequential in practice: rung (ii) fired in Experiment 6 and forced the reduction to two experts, rung 1 fell in Experiment 7 and removed the expert graph from the claim set, rung 2 fired in Experiment 8 and re-scoped the program to abstention only, and the terminal rung fired in Experiment 10. Artifacts were pinned by SHA-256 manifest, and gate evaluation required a fresh-process reload of the pinned checkpoint rather than reuse of in-memory state, a discipline tightened after a stale-checkpoint, missing-manifest-check defect surfaced in Experiment 8. Deviations from a preregistered plan were permitted only if disclosed in the run report. At least eight were disclosed across the program, including a BF16-precision change on Turing hardware, the declared deltas between Experiment 4's replanned dual-T4 run and its crashed A100-MIG original, the cross-session salvage resume in Experiment 10, and, in the terminal run, a shipped corpus of roughly 13.8k rows against a planned 25k and a masked-oversampling floor recorded at 0.37 against a preregistered $\geq$0.50.

The honest calendar is short. Dated artifacts span 2026-08-17 to 2026-09-05, twenty days; the CPU structural diagnostic carries no recorded date and may predate this window. Within it the program ran roughly twenty GPU sessions, almost all on Kaggle dual-T4 instances: an estimated 55–65 wall-clock session-hours (a dual-T4 session trains both seeds concurrently, so device-hours are roughly 110–130), plus about 0.15 A100-MIG hours before that machine crashed at step 896. Approximately 32 of those hours carry explicit accounting; the remainder is estimated. Across the window the program ran roughly 284 preregistered gate evaluations (the terminal run's two go/no-go firings excluded as retracted-vacuous) and recorded roughly 89 failures. No experiment passed its full battery: zero full-battery passes in ten experiments. The matched plain-LoRA control, which by then was gated behind a full pass, was consequently never authorized and never ran (Section 8).

Table 2 compresses the record. One detail deserves note: the first live channel-capability gate pass in program history, the terminal run's warm gate at 2/2, arrived on 2026-09-05, the same day the run was killed after the novelty audit (Section 6) returned no surviving claims.

Table 2: Program timeline, one row per run. Dates are 2026; all dual-T4 sessions ran on Kaggle; GPU-h are wall-clock session hours (a dual-T4 session trains both seeds concurrently, so device-hours are roughly double); "n/r" means not recorded and ∼ marks estimates. Gate tallies aggregate both seeds; red marks a battery that did not fully pass. Experiment 10's tally is the effective count after one canary failure was certified out-of-band as a resume artifact.

Exp	Dates	Hardware	∼GPU-h	Gates	Outcome
—	n/r	CPU	0	none	Structural diagnostic: plumbing check; simpler controls tied or exceeded it.
1	n/r (≤08-19)	2×T4	n/r	0/2	Trained stably; coverage zero; checkpoint lost to session restart.
2	08-19	2×T4	n/r	0/2	Teacher-forced calibration did not transfer; coverage 12.5%.
3	08-19	2×T4	n/r	0/2	Seed 29 failed only calibration; routing 98.4% one-expert.
4	08-30	MIG → 2×T4	∼3.6	30/36	Calibration failure fixed and replicated; three routing sub-gates failed.
5	08-30–31	2×T4	∼6	26/36	First routing-load pass (seed 29); seed 17 collapsed.
6	08-31	2×T4	∼6	28/36	Variance fixes rescued degeneration; routing collapsed both directions.
7	08-31–09-01	2×T4	∼5	25/36	Two experts bought routing, cost content; expert graph exited.
8	09-01–02	2×T4	∼12–13	20/40	Fan-in lost decisively to self-consistency; re-scoped to abstention.
9	09-02	2×T4	≈8	34/48	Every repair vindicated; every repaired mechanism dead.
10	09-02–03	2×T4	≈9.6	30/44	Masked accuracy 0.000 both seeds; terminal rung fired.
—	09-03–05	2×T4	∼6–8	2/2 warm	Dense-supervision run: first warm-gate pass; killed after the novelty audit.

Preregistration bought exactly one thing, and it was worth having: it prevented post-hoc spin. The clearest instance came after Experiment 4, the program's nearest miss, where 15 of 18 sub-gates passed on each seed and both seeds failed the same three routing sub-gates, with one expert taking 91–94% of traffic. The battery was not amended to excise the failing sub-gates and declare a pass; the routing failure stood as recorded, and Experiment 5 was designed as a routing redesign under a fresh preregistered battery. The same discipline kept the ladder rungs binding and kept every deviation on the record. What preregistration

did not buy was measurement validity. A gate battery certifies the numbers a harness emits about the artifact the harness presents; it offers no protection when the harness measures a different system than the plan describes. The program discovered five headline instrument-defect classes and roughly thirty individually documented defects, and the post-termination audit of the terminal run found an expert graph called zero times, routing implemented as a gold-label lookup, and two of fourteen gates hardcoded to pass. No preregistered threshold could have caught any of these. Section 7.1 treats this failure mode in full.

4. The Experimental Record

The program ran ten preregistered experiments, preceded by one CPU-scale structural diagnostic. Every experiment trained two independent seeds, 17 and 29, one per GPU; every experiment was judged against a preregistered gate battery evaluated by a fresh-process audit of the saved checkpoint [46]; and no experiment passed its complete battery. Read end to end, the record is not a sequence of architecture results. It alternates between two kinds of work: repairing the measurement instrument until a null result could be believed, and, once a measurement was trustworthy, falsifying the mechanism it measured. Experiments 1–5 (Section 4.1) built the instrument — semantic graders, an honest publication boundary, on-policy calibration — and established that expert routing was never causally live. Experiments 6–8 (Section 4.2) measured training variance, cut the routing machinery to its minimum viable form, and exposed further instrument defects. Experiments 9–10 (Section 4.3) confirmed the repairs and recorded the central read–write dissociation.

Three reporting conventions hold throughout. First, each experiment is reported in a fixed shape: the preregistered question, the headline numbers, the gate outcome, the instrument defect the run exposed, and the change the next revision made. Second, sample sizes are small and are stated: behavioural probe accuracy rests on $n = 32$ audit anchors per seed ($n = 16$ in Experiment 2), with $n = 8$ per grader family; the causal-liveness routing intervention rests on $n = 8$ records per seed; every experiment has exactly two seeds. The record reports no confidence intervals, a deviation from its own statistical protocol that we flag here rather than repair post hoc. Third, gate outcomes are compressed inline and in Table 3; the full per-experiment gate tables are in Appendix A.

4.1 Experiments 1–5: building an instrument that could measure anything at all

Table 3: Experiments 1–5 at a glance: preregistered gates passed per seed and the defining failure of each run. All five complete batteries failed on both seeds. Sub-gate arithmetic ($n/18$) exists only from Experiment 4, when the battery was frozen at 18 named sub-gates. Probe metrics rest on $n = 32$ audit anchors per seed ($n = 16$ in Experiment 2; $n = 8$ per grader family); the routing intervention rests on $n = 8$ records per seed.

Experiment	Seed 17	Seed 29	Defining failure
Structural diagnostic	no battery (seed 7 only)		0/320 sequences committed; verifies plumbing only
1 Substrate transplant	fail	fail	coverage 0; graders and labels defective; checkpoint lost
2 Graded evaluation	fail	fail	calibration-to-inference mismatch: 0.918 teacher-forced balanced accuracy beside 0.125 generated coverage
3 Publication boundary	fail	fail	negative rejection 0.583 (gate ≥ 0.70); degenerate thresholds; routing gate too weak to gate
4 On-policy calibration	15/18	15/18	routing cluster: load 0.063/0.094, entropy 0.130/0.174, intervention 0.0035/0.0008
5 Routing redesign	10/18	16/18	seed split: s17 100% one-expert with leak 0.266; s29 load 0.418 and entropy 0.570, intervention still 0.0073

The structural diagnostic. Before any GPU run, a 41,542-parameter CPU network verified that the code paths execute at all. Each configuration trained for 300 updates on seed 7 and was evaluated on the same 320 generated sequences (2,560 tokens). The full configuration scored 601/2,560 candidate tokens (23.48%) against 470/2,560 (18.36%) for a separately trained no-workspace control — a 5.12-point difference with no uncertainty attached, on a network lacking 12 of the specification's mechanisms; several simpler controls tied or slightly exceeded the full model. No complete sequence was correct (0/320)

and none was committed (0/320), so coverage was zero and the false-commit rate conditional on commitment was undefined: the first appearance of the commitment pathology that recurs through the program. The diagnostic verifies plumbing only. Neither its numbers nor Experiment 1's have on-disk provenance; Experiments 2–5 are reproducible from shipped JSON.

Experiment 1 `v5.2`: substrate transplant. Question: can a frozen Qwen3-0.6B substrate host the dual-mode workspace stably? The transplant (669,060,555 total, 73,010,635 trainable parameters) trained 896 microsteps per seed with zero skipped updates. The behavioural numbers that exist are best-of-two-seeds — seed 29's were never recorded: candidate token F1 0.469 against a frozen-base 0.312 (no superiority claim supported), quality pass 0.75, commitment coverage 0 (clean candidates ranked above corrupt ones, yet fixed thresholds withheld every answer), and mean lane-summary cosine 0.947 — the lanes nearly identical despite different briefs. Both capability gates failed. The run's value was the defects it surfaced: exact-phrase graders rejected semantically correct answers; missing-evidence anchors labelled safe abstention "do not publish", so a correct uncertainty statement could never be published; thresholds were never calibrated; and the 4,300-episode corpus had zero independently adjudicated episodes, 4,277 publish labels, and 99.5% of lane routes on one generic path. The checkpoint was then lost to a session restart — a reproducibility failure converted into permanent protocol: saved versions, SHA-256 sidecars, fresh-process reload. Next: semantic graders, safe abstention as a publishable answer, fitted thresholds, and a 32-generation audit per seed.

Experiment 2 `v5.3`: graded evaluation. Question: do semantic graders, safe-abstention publication, and threshold fitting produce honest measurements? Both seeds failed the battery: probe semantic accuracy 0.625 and 0.563 against a 0.75 gate, safe-abstention accuracy 0 versus 1.000 across seeds, commitment coverage 0.125 on both — four of 32 candidates published per seed. The defining defect is the calibration-to-inference mismatch. On teacher-forced references and paired corruptions, seed 29's calibration looked strong: 0.906 clean acceptance, 0.930 corrupt rejection, 0.918 balanced accuracy. On the model's own freely generated candidates, the same machinery published 12.5%. The mechanism was specific: correct missing-evidence answers carried commitment probabilities near 0.95 but private denoising-error probabilities near 0.74, above the fitted 0.590 threshold, and were withheld. The audit caught this only because both numbers sat in one report; a selective-prediction system calibrated on anything but its own emissions is calibrated for a different decision problem. One fix did land: lane cosine fell from 0.947 to 0.496/0.331. Next (one boundary, two changes): a candidate-level commitment module with three logits scored on the frozen candidate snapshot, and thresholds fitted on generated validation emissions.

Experiment 3 `v5.4`: the publication boundary. Question: does fixing the publication boundary transfer calibration to generated candidates? Partly. Seed 29's generated validation emissions were semantically correct at 0.875 and its fitted thresholds accepted positives at 0.839, but negative rejection was 0.583 against a 0.70 gate (balanced accuracy 0.711), and the fitted operating point was degenerate: commitment threshold 0 with risk and verifier thresholds near 10^{-3} — overlapping, poorly scaled policy scores, not a deployable boundary. Both seeds failed the battery. The run also exposed a gate too weak to gate: seed 17 sent 98.4% of audited routing decisions through one expert, yet the then-current "multiple routes" criterion (any second route above one percent) would have counted this as specialist use. Next: an on-policy head phase trained on the model's own generated emissions, and three preregistered routing gates — second-route load ≥ 0.10, normalized route entropy ≥ 0.25, and a causal intervention (pin all routing to the least-used expert) that must move mean absolute commitment probability by ≥ 0.01.

Experiment 4 `v5.5`: on-policy calibration, two-seed replication. Question: do on-policy heads separate the model's own emissions, and does routing survive causal gates on independent seeds? The calibration fix worked and replicated: negative rejection rose from 0.583 to 0.889 and 0.917 on independently trained seeds, balanced accuracy near 0.95 on both — held out relative to the 288 head-phase training records, though the same 64 generated validation emissions also served the threshold search, and no test-split calibration number was ever produced in Experiments 1–5. Both seeds passed 15 of 18 sub-gates and failed the same three, the routing cluster: second-route load 0.063/0.094 (gate 0.10), normalized entropy 0.130/0.174 (gate 0.25), causal intervention 0.0035/0.0008 (gate 0.01). The mechanism diagnosis is the most reusable finding of the early record: the marginal-entropy regularizer did its literal job, driving the

KL divergence of mean routing probabilities to uniform down to 0.01–0.03, while expert 0 kept 93.8% and 90.6% of hard decisions and experts 2–5 received zero decisions on both seeds. The regularizer moved the probabilities, not the choices — and near-uniform soft mixing feeds every expert nearly identical gradients, homogenizing them. A post-termination code audit sharpened this further: training-time routes were sampled multinomially while evaluation used argmax, a rule under which homogenization is predicted regardless of the balance objective (Section 7.1). Two qualifications the original write-up dropped: the intervention rests on $n = 8$ records per seed, and the policy heads were saturated at the rails (5×10^{-5}/0.9999 after 400 head-phase epochs), which compresses intervention deltas, so causal deadness and head saturation are confounded in this measurement. The fitted commitment threshold also degenerated to 0.0 on both seeds, the second recurrence of a correlated-label pathology fixed structurally only four experiments later. The external-benchmark step was not authorized because the full gate failed; the internal record explicitly refused a post-hoc amendment — the calibration machinery, the part benchmarks would showcase, passed everywhere, but "the rule is the rule". Next: a Switch-style balance loss on hard assignments [12, 9] strengthened 0.01 → 0.05, a pairwise expert-output diversity penalty (weight 0.05), de-saturated heads (400 → 120 epochs, 0.05 label smoothing), and a widened multi-domain data boundary — three simultaneous mechanism changes plus new data, a design confound Experiment 6 later removed with a two-flag follow-up.

Experiment 5 `v5.6`: routing redesign on verified multi-domain data. Question: does hard-assignment load balancing produce causally live multi-expert routing on execution-verified multi-domain data? The corpus grew to 7,498/945/841 rows, including 1,699 execution-verified benchmark episodes (166 references dropped for failing their own official checks); the 18 gate keys were byte-identical to Experiment 4's. For the first time the seeds did not fail the same way. Seed 29 delivered the record's first pass of a routing-load gate: three experts at loads 0.16/0.42/0.42, second-route load 0.418 (4.2× its gate), normalized entropy 0.570 (2.3×) — thresholds no prior experiment had exceeded 0.094 and 0.174 on — while the causal intervention improved to 0.0073 against the 0.01 gate and still failed ($n = 8$ records); it failed 2 of 18 gates. Seed 17, under the identical configuration and data, collapsed to 100% one-expert routing, 26.6% prompt leakage, and probe accuracy 0.281 (0.875 in Experiment 4), failing 8 of 18 — while its plain causal path stayed exactly as healthy as seed 29's (token F1 0.588 versus 0.589) and every training-time signal looked normal: calibration 0.818/1.000, non-degenerate thresholds, smooth losses. The collapse was visible only at generation time. Its texture — numerically correct but format-broken candidates such as "Human: 80329 + 3996 = 84325." against the reference "The result is 84325.", with only 3 of the 9,284 corpus rows containing chat markers — points at a degenerate workspace prefix admitting the base model's dialogue prior, not data contamination. The run also exposed a structurally dead loss term: the diversity penalty read 0.0002–0.0018 for the entire run, because zero-initialized expert output projections keep every expert delta at zero, leaving nothing to decorrelate — the first member of a defect class (losses and gates dead at initialization) that recurs twice more (Section 7). Seed 29's diversity is therefore attributable to the strengthened balance term plus optimization noise, an attribution no run isolated. The preregistered fallback (cut experts 6 → 2) was not triggered: its condition required both seeds to fail the routing gates, and seed 29 passed two of three. The next revision changed exactly two variables with the data byte-identical — nonzero expert-output initialization (std 0.01) and a rebalanced 128-record head phase; Section 4.2 records the split verdict, in which the initialization cured the degeneration while routing collapsed again.

Two closing cautions on this stretch of the record. The token-F1 gate, the closest thing to a baseline in Experiments 1–5, is a non-inferiority test against the frozen *untrained* base: Experiment 5's seed 17 passed it (0.324 against a 0.246 bar) while leaking prompts on 26.6% of outputs and scoring 0.000 on two of four grader families, so it is not sensitive to total channel failure — which is exactly why the unrun matched control matters (Section 8). And every behavioural conclusion above rests on 32 audit rows per seed, where a single row moves a metric by 0.031 and gates were decided on margins as small as 0.010.

4.2 Experiments 6–8: minimum-viable routing and the fan-in defeat

Experiments 6–8 walked the expert graph down from six experts to none. Experiment 6 asked whether Experiment 5's two seed-specific failures yield to one-variable fixes; Experiment 7 executed the preregistered reduction to a minimal two-expert configuration; Experiment 8 abandoned routing and asked whether the calibrated heads, which until then had only braked, could steer. Table 4 gives the per-seed record; full

gate tables are in Appendix A.

Experiment 6: targeted variance fixes. Exactly two flags changed on the otherwise frozen configuration, both seeds (17 and 29) re-run for the full 4,224 microsteps with zero skipped updates. First, the expert output projections were initialised at standard deviation 0.01 instead of zero, so that the expert-output diversity penalty — provably inert throughout Experiment 5 — exerted pressure from step one; this lever was aimed at seed 17's generation degeneration. Second, the head phase grew from 96 to 128 records with per-family round-robin balancing, aimed at seed 29's confident-false-rejection cluster. The initialisation fix rescued the degeneration cluster completely: seed 17's prompt leakage fell from 0.266 to 0.0, probe semantic accuracy rose from 0.281 to 0.781, and probe commit accuracy from 0.656 to 0.781, its first pass on that gate. It did not rescue routing: seed 17's second-route load and normalized route entropy both read exactly 0.000 against gates of 0.10 and 0.25 — collapse to a single expert for the second time under a second initialisation. Degeneration and collapse were therefore not one failure mode; the fix decoupled them. The balance fix was nullified at collection time: the round-robin balanced the *requests*, not the *valid records*. Of 128 requested head-phase emissions, 85–87 were graded invalid, leaving 41–43 valid positives against roughly 342 negatives — the same ∼1:8 imbalance the fix targeted — and seed 29's probe commit accuracy stayed at 0.625 against the 0.70 gate, unchanged to the third decimal. Seed 29's routing gates passed, but the margins collapsed from 4.2× and 2.3× to 1.1× and 1.2× (load 0.418 → 0.113, entropy 0.570 → 0.306): routing diversity is fragile, not merely seed-dependent. The routing intervention gate failed on both seeds ($|\Delta\,\text{commit}|$ of 0.0012 and 0.0031 against ≥ 0.01), the fourth consecutive experiment to fail it; causal routing liveness stood at 0-for-8 seed-runs. Four of 18 gates failed per seed. The fallback ladder's second rung fired, preregistering the reduction to two experts, a validity-gated head phase (floor of 16 grader-verified valid positives per family, capped at 512 attempts), a report-only generation-time canary, and a threshold non-degeneracy diagnostic — with an escalation commitment recorded in advance: if either seed still collapsed at two experts, the expert graph would be removed from the small-scale claim entirely.

Experiment 7: minimum-viable routing. The reduction executed. Seed 17 collapsed to a single expert for the third time under a third configuration, and the fitted commitment threshold was degenerate on both seeds (recurrences five and six). Seed 29 posted the healthiest routing in the program — second-route load 0.273, normalized route entropy 0.846, and lane summary cosine 0.829, the first distinct-lane pass in seven experiments — yet the causal test still failed on both seeds (interventions 0.0082 and 0.0037 against ≥ 0.01), bringing routing liveness to 0-for-10 seed-runs. What the reduction bought in routing it paid for in content: probe semantic accuracy fell from 0.781 to 0.563 on seed 17 and from 0.750 to 0.469 on seed 29 — a within-architecture loss of 0.19–0.28, with the numeric and unit probes at 0.25 and 0.125. The reduction was a routing/content trade, not a fix: the six-expert bank had been carrying content capacity even while failing every causal-routing gate. One repair did land. With the validity-gated head phase delivering 78 of 512 and 77 of 448 balanced valid positives, seed 17's probe commit accuracy reached 0.844, the best measured in the program — direct evidence that Experiment 6's head-phase imbalance was causal for the confident-false-rejection cluster. The batteries failed 6 of 18 and 5 of 18 gates, the preregistered escalation fired, and the routed expert graph exited the small-scale claim, standing in the record as proposal plus negative evidence: three collapses under three configurations, and causal liveness 0-for-10. The canary's first finding was about the canary: anchor-stratified by construction, it showed single-expert routing at every boundary on both seeds — including seed 29, whose audit measured 0.273 second-route load — because route diversity lived in non-anchor domains the canary never sampled.

Experiment 8: verified fan-in. The workspace-only successor (one expert, two lanes, freed budget returned to LoRA sidecars) made the heads a selector: one workspace pass decoded four candidates (greedy plus temperatures 0.7, 0.9, and 1.1), the heads scored each, and the best was published if it cleared the fitted rule. The declared headline gate: fan-in beats self-consistency [45] at a matched four-candidate budget on the same rows. The battery grew from 15 to 20 gates; audits ran 64 rows per seed, 36 with the authorised grader and 33 in the baseline denominator. The calibration machinery worked — validation balanced accuracy 0.884 and 0.877, fitted publish thresholds 0.578 and 0.689, ending the six-run streak of degenerate thresholds — and the headline failed decisively in the same direction on both seeds: fan-in 0.455 against self-consistency 0.758 (seed 17) and 0.515 against 0.818 (seed 29). The diagnosis is exact: the selector was near-oracle and the generation channel starved it. Seed 17's publish-score selection reached its

own oracle exactly (0.417 against an any-valid oracle of 0.417, from greedy 0.194); seed 29 recovered half the headroom (0.472 against 0.556, from 0.444). But the workspace-conditioned oracle sat roughly 30 points below plain causal greedy from the same adapter on the same rows (0.758 and 0.818); self-consistency, max-log-prob selection, and causal greedy tied because all drew from the healthy causal channel. No selection rule recovers content that was never generated.

The root cause surfaced during Experiment 9's build, and it is one line. Qwen3-0.6B-Base defines no beginning-of-sequence token; the tokenizer's `bos_token_id` resolves to the end-of-text token. Experiment 8 seeded every workspace-channel teacher-forcing target and every workspace decode with that identifier — a document boundary mid-sequence — so the model began a fresh document, which in its pretraining corpus means a fresh dialogue turn. The causal channel never used the seed and was never affected: exactly the observed damage pattern, down to the 12 of 64 seed-17 published answers that opened with dialogue scaffold while the training canary read a leak rate of 0.0 at every checkpoint. A line-level defect with an architecture-level consequence, invisible to every training-time metric because it lived only in the emission path [47]. A post-hoc headroom analysis, disclosed as not preregistered, shows the headline was also underpowered by construction: the probability that exactly one of N candidates is correct is $Np(1-p)^{N-1}$, which at the causal pool's $p \in [0.758, 0.818]$ and $N = 4$ is 0.043 down to 0.020 — one decidable row in 23 to one in 50 against a 33-row graded audit. The comparison was doubly compromised, once by the channel and once by the arithmetic.

Rung 2 of the ladder was invoked: the verified fan-in claim is dead at this scale, and the claim re-scoped to calibrated abstention alone, seed split disclosed as observed rather than averaged. That re-scope needs its own discipline. At matched coverage the heads beat mean-log-prob abstention on seed 17 (selective accuracy 0.818 against 0.636 at coverage 0.306, the first selective-prediction pass in the program) and lost on seed 29 (0.333 against 1.000 at coverage 0.083, three published rows). Coverage 0.306 of the 36-row graded set is eleven published rows, and 0.818 against 0.636 on eleven rows is nine correct against seven: the eighteen-point reading rests on eleven published rows and a two-row margin, holds on one seed of two, and is not replicated. Rungs 3 and 4 did not fire — the read-out ablation was non-null on both seeds (0.167 and 0.306) and the scalar rule non-degenerate on both — although Experiment 9's instrument corrections later revise what that non-null ablation means (Section 4.3). One procedural note stands to the run's credit: seed 17's commitment calibration and seed 29's policy-head training reproduced an earlier crashed session byte-identically, all six policy thresholds matching to full float precision — the strongest determinism evidence in the program.

Table 4: Experiments 6–8, per-seed record. Gates in parentheses; red marks preregistered gate failures. Both seeds are 17 and 29 throughout; Experiments 6 and 7 train 4,224 microsteps with zero skips.

Measure (gate)	Seed 17	Seed 29
Experiment 6: two one-variable fixes (init std 0.01; head phase $96 \to 128$*, round-robin)*		
Prompt leakage (= 0)	0.0 (was 0.266)	0.0
Probe semantic accuracy (≥ 0.75)	0.781 (was 0.281)	0.750
Probe commit accuracy (≥ 0.70)	0.781 (was 0.656)	0.625 (unchanged)
Second-route load / route entropy (≥ 0.10 / ≥ 0.25)	0.000 / 0.000	0.113 / 0.306
Routing intervention $\lvert\Delta$ commit$\rvert$ (≥ 0.01)	0.0012	0.0031
Gates failed (of 18)	4	4
Experiment 7: experts $6 \to 2$*, validity-gated head phase*		
Probe semantic accuracy (≥ 0.75)	0.563	0.469
Probe commit accuracy (≥ 0.70)	0.844	0.469
Second-route load / route entropy (≥ 0.10 / ≥ 0.25)	0.000 / 0.000	0.273 / 0.846
Routing intervention (≥ 0.01)	0.0082	0.0037
Gates failed (of 18)	6	5
Experiment 8: verified fan-in, workspace-only, 20-gate battery		
Fan-in vs self-consistency (headline: fan-in > SC)	0.455 vs 0.758	0.515 vs 0.818
Fan-in vs own greedy (≥ 0)	+0.222	+0.028
Workspace oracle vs causal greedy (report)	0.417 vs 0.758	0.556 vs 0.818
Turn-scaffold leak rate (= 0)	0.188	0.016
Fitted publish threshold (non-degenerate)	0.578	0.689
Selective accuracy, heads vs log-prob (matched coverage)	0.818 vs 0.636 (cov. 0.306)	0.333 vs 1.000 (cov. 0.083)
Gates failed (of 20)	9	11

4.3 Experiments 9–10: the repaired channel and the necessity test

Experiment 9: every repair vindicated, and the mechanism dead anyway. Experiment 9 rebuilt the emission path around the bos/eos repair (Section 4.2) and decoupled the surviving claims onto independently powered surfaces: channel parity (Claim G), verifier-weighted selection (Claim S), and conformal abstention (Claim A) [37, 40]. Both channels now shared one prompt builder and differed only by the latent prefix, making the read-out ablation exact. Every repair held: across 640 audited decodes (320 rows per seed) the turn-scaffold leak rate was 0/320 on both seeds, against Experiment 8's 0.188 and 0.016; workspace-conditioned greedy accuracy sat within 0.013 of causal on seed 17 and 0.003 above it on seed 29; token-F1 ratios were 1.015 and 1.025. Experiment 8's thirty-point channel gap was gone.

The mechanism the repair served was dead anyway. Ablating the 34 memory positions of the 42-token prefix changed graded accuracy by 0.000 on seed 17 and -0.022 on seed 29 (the ablated arm was 0.022 *more* accurate); both fail the ≥ 0.05 liveness gate. And the parity Claim G recorded was parity by irrelevance: both channels saw the full context, so the latent was redundant by construction, and a decoder ignoring it entirely would produce exactly the numbers measured. Three instrument facts from the post-session code audit discipline even this reading: the shipped ablation covered only 34 of 42 prefix positions, so the full prefix is bounded by parity itself (the causal arm is the workspace arm with the entire prefix removed); the prefix gate $\tanh(\alpha)$ moved less than 0.0008 but is disqualified as evidence, since scale-invariant RMS normalization leaves its gradient near zero whether or not the prefix is useful; and the fluency anchor's reference was the prefix-free model, so its persistent gradient on the prefix pathway pointed at changing the logits by nothing.

The remaining surfaces closed as follows. Claim S was void as preregistered: the medium difficulty bin held zero exactly-one-correct rows on both seeds, failing the headroom gate at 0.000 (weighted versus plurality self-consistency flipped two votes in 640 rows; pooled McNemar mid-p 0.125). Claim A recorded coverage 0.244 (in band) and 0.188 (0.013 below the $[0.20, 0.50]$ floor), published answers 78/78 and 60/60 correct, and a pass on the matched-coverage selective-accuracy gate on both seeds — heads 1.000 versus log-probability 0.987, and 1.000 versus 0.983 — a genuine pass the program's first write-up omitted from its Experiment 9 table, restored here. On the area metric the free baseline won on both seeds: AUGRC [43] 0.0637 versus 0.0594 and 0.0654 versus 0.0623, bootstrap win fractions 0.13 and 0.179 against a 0.90 bar (Section 5). Lane cosines failed for the eighth and ninth time (0.940, 0.975), executing the single-lane demotion. Each seed passed 17 of 24 gates, one acknowledged as vacuous: the shipped validity-floor gate checked that head training occurred, not that the floor was met (Table 17; Section 7). Experiment 9's bequest was a constraint, not a fix: make the latent structurally necessary, so that ignoring it costs loss, and preregister that check as the first gate.

Experiment 10: the necessity test. Experiment 10 executed the constraint. On masked behavior anchors the operative premise was moved out of the request the answer path reads: the decoder's input ids for the answer channel physically lack the operands, the workspace encoder alone sees the full context, and the only route from premise to answer is the latent. Training used the same asymmetry on alternating batches, so a model that ignores the latent pays cross-entropy for it during training, not only at audit. The instrument was airtight by its own checks: zero leaked premise literals in 198 masked audits (99 rows per seed, line-anchored scanner shared by trainer and audit); a causal floor of exactly 0.000 with the prefix removed, so the question is answerable only through the channel; a trained gist attribution control — a matched prefix trained on the same alternating schedule with no evidence access — reading 0.000 on both seeds; and unmasked parity intact (+0.037 and 0.000; seed 17's workspace arm at 0.508 against causal 0.471).

Against that instrument, the central reading of the program: masked workspace accuracy 0.000 on 99 rows per seed, on both seeds, while a latent-only linear probe decodes 0.297 and 0.334 of the withheld premise. The roughly 0.30 recovery itself fails the preregistered 0.50 decodability bar, and no chance floor for the probe was recorded, so these values stand against the bar alone: the latent holds partial, probe-recoverable traces, and generation through the same latent recovers none of them. This is the constructed and repaired counterpart of Lagged Coupling [1]: where that work measured in pretrained models that internal representations become readable before they become causal, this program built the channel, deleted every alternative route to the answer, repaired every instrument defect its record documents, and measured the same dissociation at its extreme point — read-out above zero, generation exactly at zero. The signature matches the published failure mode of answer-only supervision on latent reasoning tokens: CODI's own

ablation reproduces it, GSM8k accuracy collapsing from 43.7 to 24.5 when the hidden-state distillation term is removed while the latent remains present and trained [6], and Coconut documents the same class [7]. The ladder's terminal rung executed: the generative workspace mechanism is retired at this scale, now under structural necessity rather than redundancy.

Two further properties were measured, not explained away: the verifier head, correct on teacher-forced features, inverted on freshly generated candidates (margins -0.277 and -0.227, the third consecutive experiment showing this train-to-generation shift); and on-policy generation at banded difficulty produced 41 and 43 valid emissions of 512 attempts against a floor of 16 per family, so the heads were calibrated on a distribution their own generator barely produces. Table 5 gives the headline battery. On its abstention rows: the band-restricted partial AUGRC, preregistered before the run, favours the heads on both seeds, but its significance test fails on both (0.746 and 0.792 against the 0.80 bar), and on full-range AUGRC the seeds split in opposite directions, bootstrap win fractions 0.994 and 0.039; Section 5 carries the full account, including the metric, bar, and band changes between Experiments 9 and 10.

Table 5: Experiment 10 headline battery (288 audit rows per seed: 224 banded anchors, of which 99 masked, plus 64 text-to-SQL). Preregistered gate failures in red. The coverage band was widened from Experiment 9's $[0.20, 0.50]$; both readings would have passed either band. Partial and full-range AUGRC use different normalizers and are reported together; the full-range bootstrap split (0.994 / 0.039) shows the seeds disagreeing in opposite directions on the unrelaxed metric.

Measurement (gate)	Seed 17	Seed 29
Microsteps completed / skipped	3,200 / 0	3,200 / 0
Masked-row premise leaks, 99 rows (gate = 0)	0	0
Causal floor, prefix removed, masked rows	0.000	0.000
Masked workspace accuracy (terminal-rung gate > 0)	0.000	0.000
Trained gist attribution control, masked rows	0.000	0.000
Latent-only premise probe (gate ≥ 0.50)	0.297	0.334
Unmasked parity gap, workspace − causal	+0.037	0.000
Verifier margin on generated candidates (gate > 0)	−0.277	−0.227
On-policy validity floor (gate ≥ 16 per family)	41 / 512	43 / 512
Conformal coverage (band $[0.20, 0.55]$)	0.295	0.267
Published answers correct	83 / 85	77 / 77
Selective-risk UCB_{95} (gate ≤ 0.15)	0.072	0.038
Partial AUGRC, heads vs. log-prob (band $[0.05, 0.50]$)	0.0128 vs. 0.0156	0.0073 vs. 0.0114
Full-range AUGRC, heads vs. log-prob	0.12731 vs. 0.14122	0.13871 vs. 0.12646
Paired-bootstrap win fraction, partial (bar 0.80)	0.746	0.792
Paired-bootstrap win fraction, full range	0.994	0.039
Gates passed (of 22; canary artifact excluded)	15	15

The sharpened constraint. Experiment 9's lesson was necessity; Experiment 10's is that necessity is not sufficiency. Every published system that does pass content through a latent interface into a frozen decoder shares one ingredient no experiment in this program had: a dense, token-level objective through the latent itself — reconstruction or distillation — whose targets cannot be predicted without the latent's content [6, 7]. This program's ten experiments supervised the answer and hoped the latent would become useful; it did not, under redundancy or under necessity. The successor designed to test dense supervision directly was terminated at roughly 125 optimizer updates, against the 2×10^5 to 1.5×10^7 updates of published successes in this class — its warm-start gate had passed 2 of 2 at 37 updates, the first live channel-capability gate pass in the program's history, and the run was stopped on novelty grounds anyway. That question therefore closes undecidable at this budget, not falsified.

5. The Thread That Kept Passing, and What It Actually Shows

One component survived every redesign of the program: the coverage-controlled publication stack. Table 6 gives the full arc. It begins at coverage zero, spends six consecutive runs producing degenerate publish thresholds, reaches its first non-degenerate rule in Experiment 8, and lands inside a preregistered coverage band in Experiments 9 and 10. The honest headline is this: across four seed-runs and 300 published answers (Experiments 9 and 10), coverage-controlled publication was operationally flawless — Experiment 9 published 138 answers, all correct; Experiment 10 published 162 with 160 correct (83/85 and 77/77; an earlier draft misreported these totals as 160 published and 158 correct, both figures off by two, and the

arithmetic is corrected here) — and it beat the free mean-log-probability baseline by six rows in total, 298 correct against 292 at matched coverage. It never cleared its own preregistered significance bar. The headline abstention gate passed once in six seed-runs: Experiment 8, seed 17, where the heads scored 0.818 (9/11) against the baseline's 0.636 (7/11) — eleven published rows, a two-row margin.

Table 6: The abstention thread across Experiments 1–10: publish coverage (seed 17 / seed 29) and the behaviour of the publish rule. Preregistered gate failures are marked in red.

Experiment	Coverage (s17 / s29)	Publish rule
1	0	nothing published
2	0.125 / 0.125	4 of 32 rows; safe-abstention accuracy 0 / 1.000, seed-sensitive
3	—	threshold degenerate at 0
4	0.75 / 0.41 (commit)	threshold 0.0 on both seeds; heads saturated
5	—	threshold degenerate
6	—	threshold degenerate
7	—	threshold degenerate on both seeds; streak reaches six
8	0.172 / 0.078	first non-degenerate rule (0.578 / 0.689); the seed-29 strangle, safe-abstention accuracy 0.000
9	0.244 / 0.1875	split-conformal, $k = 83$; band $[0.20, 0.50]$; seed 29 misses the floor by 0.0125
10	0.2951 / 0.2674	split-conformal, $k = 58$; band $[0.20, 0.55]$; both seeds in band

The metric story must be told in full. Experiment 9's headline metric was full-range AUGRC [43]: the heads lost on both seeds (0.0637 / 0.0654 against the baseline's 0.0594 / 0.0623), with paired-bootstrap win fractions 0.13 and 0.179 against a preregistered 0.90 bar. For Experiment 10 the metric moved to partial AUGRC restricted to coverage $\in [0.05, 0.50]$ [42], the bootstrap bar dropped from 0.90 to 0.80, and the coverage band widened from $[0.20, 0.50]$ to $[0.20, 0.55]$; the model itself also changed (trainable parameters 76.63M to 132.2M). Three of the four changes favoured the hypothesis. Three facts qualify that reading, in both directions: the partial metric was preregistered before the run, with a written rationale (full-curve AUGRC on a 59%-easy pool measures the baseline's saturation, not abstention quality, and 0.80 was the pre-computed achievable bar at $n = 288$); the selective-risk upper-bound bar tightened the other way, from ≤ 0.45 to ≤ 0.15; and the metric was nonetheless selected after observing the previous run's loss on the full metric. None of it rescues the result. The relaxed bar was still missed on both seeds: partial AUGRC 0.012847 / 0.007345 for the heads against 0.015572 / 0.011351 for the baseline, bootstrap win 0.746 / 0.792, short by 0.054 and 0.008. And the full-range Experiment-10 numbers, computed, recorded in the run artifacts, and never printed in the earlier draft, go here beside them: heads 0.12731 / 0.13871 against baseline 0.14122 / 0.12646, bootstrap win fractions 0.994 and 0.039. On the unrelaxed metric the two seeds disagree in opposite directions. The two metrics also use different normalisers — full-range divides by all rows, partial by in-band weight — which alone makes 0.0128 look roughly ten times smaller than 0.127; a reader given only the partial numbers cannot reconstruct the full ones.

The flawless operating points are not evidence for the mechanism. At coverage 0.19–0.30 on a pool that is 58–59% trivially easy, any monotone confidence score yields near-perfect selective precision [38]; publishing only the easy stratum is what coverage control does. The safe-abstention showpiece makes the point exactly: text-to-SQL commit coverage of precisely 0.000 on both Experiment 9 seeds is the rule mechanically deleting a domain the model could not do at all (SQL quality 0.078 / 0.109) — correct behaviour, but a property of coverage control, not of learned commitment, risk, or verification heads.

The comparison itself was already owned. Feng et al. showed that a separately trained selection head loses to the model's own scores across coverages [41]; the same comparison, with the same conclusion, on the same model family, is published in [44]. The guard gate confirms the loss is informative rather than degenerate: Spearman correlation between the publish score and mean log-probability was 0.830 / 0.821, under the 0.95 ceiling, so the heads were not a re-derivation of likelihood — they were simply not better than it on this distribution.

One positive in this thread is defensible, and it is an engineering repair. Six consecutive runs produced degenerate thresholds, and Experiment 8's midpoint rule strangled seed 29 to coverage 0.078 with safe-abstention accuracy 0.000. Split-conformal order-statistic thresholding, $k = \lfloor (n+1)(1-c) \rfloor$ [37, 40], replaced this with banded, non-degenerate coverage: 0.244 / 0.1875 in Experiment 9, then 0.295 / 0.267 in Experiment 10, on continuous publish rules (329 and 372 distinct scores, tie mass 0.008–0.0104). Three caveats ship with it: it is a theorem doing its job under correct implementation, not an empirical discovery; it missed its band on one of four seed-runs (0.1875 against the 0.20 floor); and it was the standard method the program had simply not been using. Symmetric disclosure also requires reporting the pass the earlier

draft omitted: `selective_accuracy_matched` passed on both Experiment 9 seeds, 1.000 against 0.987 (78/78 versus 77/78) and 1.000 against 0.983 (60/60 versus 59/60).

The correct reading of this thread is a power result, not a near-miss awaiting vindication: at 288 audit rows per seed, the design could only have detected an effect roughly three times the size of the one observed. Any successor inherits an audit-set requirement in the thousands of rows, not hundreds.

6. The Novelty Audit

The program ended with a search, not an experiment. After the tenth experiment we ran a systematic prior-art audit before writing up: approximately 200 search queries across twelve independent literature-search passes, with each of the program's six architectural claims subjected to adversarial refutation and every load-bearing citation verified against its primary source. No claim survived. The mean score was 0.63 out of 5: the expert graph, the dense-supervision necessity instrument, and the methods-and-taxonomy claim each scored 1 of 5; the latent workspace, the private diffusion canvas, and calibrated publication each scored 0.25 of 5. Adversarial re-reading produced exactly one downgrade of a match grade (exact to close) and several corrections that strengthened the prior art. The audit is a literature search, not a proof of prior art; its verdicts on 2026-dated preprints were checked against primary sources, and no manufactured citation was found. Table 7 states each idea as conceived, the prior work that already contains it, the match grade, and what, if anything, survives.

Two features of the table deserve emphasis. First, the strongest matches are not paraphrases but shipped code and measured ablations. The graded blend the program conceived as its hierarchical contribution is the Residual-MoE implementation in DeepSpeed's `moe/layer.py`, which computes a learned per-token softmax blend of an always-on dense path with routed experts [13]; and the value of an always-on shared expert is not merely published but quantified, DeepSeekMoE reporting Pile loss rising from 1.808 to 2.414 when the shared expert is disabled at equal compute [15]. Second, the headline empirical finding was itself published four days before the audit ran: Lagged Coupling [1], the terminal paper of a four-study preregistered chain [2, 3, 4], documents the read-before-causal dissociation across 160M–12B models, 48 model-checkpoint cells, and four task families, with a preregistered OLMo-2 replication, against this program's two seeds at 0.6B. What remains of the program's result is stated in Section 6.1.

Production practice points the same way. The Kimi model line shares a single mixture-of-experts substrate across K2, K2.5, and K2.6 — 384 experts, 8 active per token, 1 shared — and holds it fixed while capability scales elsewhere (per its report and model cards) [21, 22, 23]. Role decomposition happens at the agent-swarm layer above the model, not inside the MoE: K2.6 coordinates up to 300 sub-agents across 4,000 steps [24], and the K3 report continues the same division of labour [25]. Industry solved role decomposition at a different level of the stack from the one this program modified.

The audit cost roughly forty minutes of automated search. Every architectural row in Table 7 rests on work published before the program's first dated artifact, so the audit required nothing the program produced; only the headline-finding row cites work that arrived during the program's final weeks. Run before Experiment 1, the same forty minutes would have returned the same verdict on all six claims before a single GPU-hour was spent. Section 9 draws the corresponding lesson.

6.1 What remains unoccupied

Three cells in the cross-product of published components are, on the audit's verified record, literally unoccupied. We state each in its narrowest defensible form.

Graded blending of role-decomposed experts. No published system blends role-in-workflow experts in learned graded (softmax) proportions inside a single LLM forward pass. MoRAgent's role gate is hard: it forces exactly one role's contribution to be non-zero per token [17]. MapCoder-Lite swaps one adapter per agent call [18]. The tier-ladder line averages simultaneously active experts with fixed uniform weights. Residual-MoE blends gradedly, but a dense path with topic-routed experts, not roles [13]. The Kimi line splits roles above the model, at the agent-swarm layer [21, 24]. Ten independent search formulations failed to find the graded role blend, and the nearest published evidence runs against it: in a task-decomposed LoRA-expert setting, hard expert selection significantly outperforms soft blending [20], and phase-level consistent expert assignment beats fine-grained token-level mixing in agentic reinforcement

Table 7: Prior-art audit of the program's architectural claims and of its headline empirical finding. Match grades: *exact* = mechanism published as conceived; *close* = published with minor differences; *adjacent* = related but distinct. No row survived as a contribution; the surviving residues are stated in their narrowest defensible form in Section 6.1.

Idea as conceived	Prior work	Match	Surviving residue
Root-preserving expert graph: routed leaves execute root-connected paths, variable depth, stage-conditioned	[10] (2018); [11] (2020); [16] (2024)	exact	A reporting norm (publish the distribution of active parameters and FLOPs; charge routing overhead to the budget), already urged by dynamic-network surveys. A methods paragraph, not a contribution.
Graded generalist/specialist blend: always-on generalist softmax-mixed with routed specialists in learned per-token proportions	[13] (2022); the shipped `moe/layer.py` computes a learned per-token softmax blend of an always-on dense path with routed experts	exact	An ablation-shaped gap: no one has made the always-on component a deliberately broader-competence generalist and studied that asymmetry as the object of interest.
Tiered generalist→polymath→specialist ladder, all tiers simultaneously active, graded blend	[8] (1994); [14] (2022)	close	No published tiers are ordered by breadth of competence, and no decoder LM blends three breadth-ordered tiers with learned graded weights; a training-and-labelling choice on a published structure.
Role-split experts (reasoner/ executor/summarizer) inside one backbone	[17] (2025); [18] (2025)	close	The hard-gated version is published; graded blending of roles in one pass is not (Section 6.1).
Multi-timescale isolated lanes: one weight-tied transition at three rates, barrier-gated commit	[26] (2014); [27] (2017); [28] (2025)	exact	The literal conjunction is unpublished; its only architectural delta from published multi-rate recurrence is tying the transition across rates, a capacity reduction. Nothing in the literature predicts the conjunction is load-bearing.
Calibrated atomic publication: trained heads decide publish-vs-fallback under a coverage target	[37] (2005); [38] (2017); [41] (2023)	exact	The separate-selection-head design was already published as a negative [41]; the residue is bookkeeping (a metric variant reported on a new substrate) with no decision consequence.
Shared-expert isolation: an always-executed shared expert beside routed experts	[15] (2024)	exact	None; the always-on component and its ablation are published.
The headline empirical finding: latent content readable by a probe while causally unusable in generation	[1] (2026), posted four days before the audit ran; chain [2, 3, 4] (2026)	exact	An interventional test bench for the measured lag (Section 6.1); replication and extension, not discovery.

learning [19]. The cell is unoccupied — and this program's preregistered causal-liveness gate failed on every seed-run that tested it (0 of 8 informative seed-runs), so we hold no evidence that occupying it does anything.

Dense latent self-distillation with physical premise deletion. No published work combines dense latent self-distillation in the style of CODI [6] with physical deletion of the premise from the student's input ids — necessity enforced in the input itself, not by attention masking. Lagged Coupling measures the lag between readable and causal representations in pretrained models [1]; this combination is the interventional test bench for that finding: a channel the decoder must use, because the content exists nowhere else in its input. The nearest neighbours occupy adjacent coordinates without the physical-deletion component: Cartridges removes the raw corpus at inference and distils from generated traces [30], and test-time context distillation aligns teacher and student hidden states under a context-withholding asymmetry [29]. The cell is unoccupied — and the one experiment that ran the combination terminated undecided at roughly 125 optimizer updates (2,000 row-presentations), where published successes in this class train with 2×10^5 to 1.5×10^7 row-presentations; the question the bench was built to answer remains open at that budget.

A private diffusion canvas with a single conformal exit. No published system pairs a categorical text-diffusion generator [33] refining a private canvas with a whole-answer split-conformal publish-or-fixed-fallback decision [37, 40] as the canvas's only exit. This is an unfilled cell in a cross-product, not a gap in knowledge: the split-conformal rule consumes only i.i.d. calibration draws and a scalar score, and is indifferent to the generator behind it, so filling the cell requires no new theory, instrument, or failure mode.

The cell is unoccupied — and this delta is design-only: on the terminal experiment's executed path the diffusion refiner was dead code (Section 7.1), the preregistered ablation isolating the diffusion contribution was never implemented, and the composed system was never exercised in any of the ten experiments.

These three statements are the audit's complete residue. An unoccupied cell is a fact about the literature, not a contribution; it becomes one only when an effect is measured in it, and none was.

7. Instrument Validity

In bolt-on-module research on frozen decoders, most nulls are uninterpretable until the instrument that produced them is certified: this program documented roughly thirty individual defects in five headline classes, and several of its most confident readings turned out to be readings of its own instruments. The full defect record sits in the experimental record (Section 4) and the gate tables of Appendix A; Table 8 lists only the classes whose failure signatures we could not find documented elsewhere, each with its mechanism, what it invalidated, and the detector that now guards it.

Table 8 is bounded on both sides. The base shapes of several other classes in the program's record are already documented: right-padding hazards for decoder-only generation carry a runtime warning in standard tooling, host-dependent certification tests are ordinary device-agnostic CI practice, parameters-that-never-update checks exist as torchtest-style suites, and silent wrong behaviour arising from defaults and misused APIs has an empirical taxonomy [47]. This paper claims the named classes above and the certification discipline attached to them — populations certified rather than thresholds alone, movement floors rather than step-0 liveness alone, the shipped predicate audited rather than the plan prose — not the genre of failure catalogues. The record is one codebase in one program, so it supports no prevalence claim.

Two worked examples carry most of the section's weight. Experiment 9's declared liveness telemetry was a single learned gate $\tanh(\alpha)$, initialised at 0.05, scaling the whole 42-token latent prefix and logged at every evaluation boundary. The first operation the frozen decoder applies to the gated embeddings is root-mean-square normalisation, which is scale-invariant: the attention-visible content of the prefix is independent of α for any positive value, and the loss gradient on the gate through that dominant branch is approximately zero whether or not the prefix is useful. The gate moved less than 0.0008 over the full run on both seeds (0.0500–0.0507 and 0.0498–0.0504). That flat trace had been read as "training rejected the prefix"; it is equally consistent with a channel that is open and ignored, so it is uninterpretable in both directions — retained as telemetry, disqualified as evidence. The class has a second-order version. The successor's replacement monitor, a "prefix attention mass" computed from the gate values, is a data-independent bijection of the gate: under a frozen gate it passes at initialisation forever. It was killed before any run consumed it and replaced by quantities the data can move — a measured contribution ratio with a floor of 0.02 at step 600, a per-layer gate-gradient EMA floor of 10^{-7}, and a movement floor of 0.005 at step 1200.

The second example is channel-selective damage. The base model defines no beginning-of-sequence token, so the tokenizer's `bos_token_id` resolves to the end-of-text token. Experiment 8 seeded every workspace-channel teacher-forcing target and every workspace decode with that identifier — a document boundary mid-sequence — and the model reasonably began a fresh document, which in its pretraining corpus means a fresh dialogue turn. The workspace channel was poisoned: turn-scaffold leak 0.188 and 0.016 by seed, and a workspace pool oracle roughly thirty points below plain causal greedy decoding from the same adapter on the same rows (0.417/0.556 against 0.758/0.818). The causal channel never used the seed and was untouched. The training canary was blind: it read leak 0.0 at every checkpoint on both seeds, because the pathology lived only in the audit emission path, which the canary never exercised. The repair was confirmed at scale in Experiment 9 — leak 0 of 320 rows per seed, channel parity $+0.013/-0.003$. A defect confined to one code path manufactures a clean-looking sibling channel and a green monitor at the same time.

7.1 What the code was actually measuring

Certified instruments are not sufficient; the code they instrument must also be executing. After termination we audited the final dense-supervision run's shipped bundle for liveness — which modules ran on the launched path, and which gates could read false — with every measurement re-executed by a second pass.

Table 8: Instrument-defect classes the program documented and could not find named elsewhere. One row per class; entries abridged from the experimental record (full gate tables in Appendix A). Red marks violated preregistered floors.

Class	Mechanism	What it invalidated	Detector
Scale-invariant pre-norm gate	A learned scalar gate on an injected prefix feeds a pre-norm decoder; RMSNorm is scale-invariant, so $\partial L/\partial\alpha \approx 0$ whether or not the prefix is useful, and the logged trace is evidence-free.	Exp. 9's gate telemetry in both directions (gate moved < 0.0008 over the full run, both seeds).	Structural check for a norm between gate and first consumer; gradient contrast between known-useful and known-useless prefixes; in-run movement floors.
Schedule and telemetry-cadence aliasing	A period-4 family rotation aliased against power-of-2 loss alternation and a 50-step log cadence ($50 \equiv 2 \bmod 4$): each dense loss would have trained on half the families, and every logged row came from the same two.	The final dense-supervision run's loss schedule (caught pre-launch) and a telemetry-based mechanism diagnosis (retracted).	Offline simulation of the full schedule with per-family, per-loss coverage assertions; stratum-stamped telemetry; cadences coprime to curriculum periods.
Rail-degenerate joint thresholds	Policy heads trained on labels anti-correlated by construction let the joint sweep pin one threshold to the 0.0 rail; the scalar successor's midpoint threshold strangled coverage to 0.078.	Every three-head calibrated-commitment claim through Exp. 7 (six recurrences, Exps. 4–7); Exp. 8 seed 29's operating point.	Threshold non-degeneracy bound $[0.02, 0.98]$; label-correlation audit before head training; split-conformal order statistic with an explicit coverage floor [37, 40].
Gate satisfiability and headroom	Pass conditions arithmetically unreachable or unpowered: selector-vs-plurality headroom $Np(1-p)^{N-1}$ allowed one decisive row per 23–50 against a 33-row audit; a growth clause $\mathrm{em}_{600} \geq 2\,\mathrm{em}_{200}$ is unsatisfiable once $\mathrm{em}_{200} > 0.5$.	Exp. 8's headline comparison; one launch of the final run, aborted at step 600; Exp. 10's abstention design, powered only for $\sim$3$\times$ the observed effect.	Arithmetic and power audit of every gate before the run; preregistered headroom gates that can declare a claim void; unit tests of tripwires on passing values.
Partial ablation	An ablation implemented as a tensor slice removed 34 of 42 prefix tokens, leaving 8 gated synthesis tokens in both arms, so the measured deltas silently changed referent.	Exp. 9's inertness deltas ($0.000/-0.022$) as full-prefix statements; rescued only by the exact channel-parity arm (within 0.013).	Build the control arm as the treatment arm minus the entire intervention; a unit test counting positions in both arms; gate-closed equivalence tests.
Canary design	Training canaries sampled the wrong stratum, exercised the wrong code path, or read a resumed log, and stayed green through the pathologies they were installed to catch.	Exp. 7's routing readings (canary saw single-expert routing while the audit measured 0.273 second-route load); Exp. 8's channel certification; one gate reading per seed in the salvage sessions.	Stratify over the domains where the pathology can live; exercise the exact claimed emission path; validate report-only against a known-bad run before gating; treat log provenance as part of the instrument.
Vacuous selectors	A go/no-go metric selected rows by a key the normaliser never emits: the population was always empty, the 0.0 default fired, and the same read silently disabled the masked-oversampling floor.	The final run's preregistered abort (retracted: untested, not falsified) and its step-1200 checkpoint, trained at masked share 0.380/0.372 against a 0.50 floor.	Empty tripwire populations raise instead of returning defaults; a blocking selector-liveness preflight before GPU hours; schema regression tests pinned to the emitted schema.
Prereg-vs-implementation fidelity	The shipped gate predicate drifts from the plan text: a gate described as enforcing a validity floor required only that head training occurred, and an authorised control was absent from the notebook.	The reading of every Exp. 9 head result (the 16-per-family floor was unmet on both seeds); the plain-LoRA control's deferral, again.	Executable preregistration inside the sha-verified bundle as the governing document; a diff of plan predicates against shipped gate code; regression tests encoding the plan predicate verbatim.

The findings, in decreasing order of structural weight:

- The only root-connected expert-graph class in the codebase received zero forward calls on any launched path (a monkeypatched counter recorded 0, against 8 for the flat expert list); at the launched single-expert configuration it degenerates structurally, and its seven parameter tensors sat `requires_grad=True` with gradient `None`, padding the optimiser state and the disclosed trainable count.
- "Routing" as shipped was a deterministic gold-label lookup, family to expert, `(0, 0, 1, 1)`, indexed by the row's gold family label in the collator; the learned router that exists is detached by construction, trained only through an auxiliary cross-entropy at weight 0.1, and consulted only in a diagnostic arm.
- In the routed experiments, training-time routes were sampled *multinomially* while evaluation used argmax — a rule under which every expert receives roughly $1/N$ of training traffic regardless of what the router learns, so experts converge toward interchangeable functions. Under it the Switch-style balance loss reduces to $N \sum_i p_i^2$, mathematically incapable of penalising argmax concentration, and it sat at its uniform optimum (readings 1.006–1.059, where 1.0 is uniform) through episodes of 100% single-expert collapse. The routing gates measured the sampler, not the router; the causal-liveness null is a predicted consequence of the training rule, and no run report disclosed the sampler.
- The diffusion refiner was unreachable dead code on the executed path: zero call sites in the entire audit binary, while the notebook still passed refinement and diffusion-step flags that only validate configuration, size never-read buffers, and serialise into checkpoints. (Through Experiment 10 the reverse schedule genuinely executed — $[4, 3, 2, 1]$ on 288/288 audit rows per seed — but its preregistered isolating control was never implemented in any of seven audit binaries.)
- Of the twelve gate booleans in the production audit caller, two could not read false: `infrastructure_ok` is hardcoded to `True`, making the ladder's rung 0 unreachable from the audit path, and `zero_skipped_updates` is true by construction. The two genuinely load-bearing channel gates were never evaluated once.
- The binding abstention rung was structurally unable to fail: it resolves only to *confirmed* (pooled e-process wealth ≥ 20.0) or *open*, never *failed* — standard for an e-process, but it means this run could not have falsified the abstention claim.
- The `-causal-control` flag — the "plain LoRA" control — was never passed by the notebook, and as written it is not plain: its freeze predicate spares everything under `backbone.layers`, which is where the experts and the prefix attention gates live, so the control would have trained them.
- 57.3% of the disclosed trainable parameters (71,485,512 of 124,672,588) received zero gradient while being registered with the optimiser and serialised into every checkpoint; the disclosed count overstates the trained model by more than 2×, contaminating any parameter-matched-control argument.
- The reconstruction term the plan said was removed was still live at weight 0.3 in the shipped trainer: the claimed swap to pure distillation was a stack, not a swap.

The lesson of this section is one sentence: preregistration [46] constrains what you ask; only liveness certification constrains what you measure — and the two must be audited against each other, because the gaps between plan text and shipped predicate (the fidelity row of Table 8) are exactly where a program loses track of which system it is measuring.

8. Limitations

The decisive control never ran. Every comparison in this paper is against the frozen untrained base, or against training-free baselines drawn from the same trained adapter. The matched control — plain LoRA on the same frozen Qwen3-0.6B substrate, the same 3,200 steps, data mixture, audit rows, graders, and seeds 17 and 29, at the same trainable-parameter budget, with no workspace, lanes, latents, or fan-in — was identified on 2026-08-30 as "the decisive missing experiment", costs about four hours at the measured 6.75 s per step, and never ran. Until it does, no observed cost or gain anywhere in this record can be attributed to the architecture rather than to finetuning at all: as the program's own audit put it, the ten nulls answer "worse than what?" with "worse than nothing measured".

How it came never to run is a process error, and we disclose it as one. The control began unconditional: the plan for Experiment 5 (2026-08-30) stated that it "runs regardless of [the main session's] outcome: no capability claim is valid without it". On 2026-08-31, after a bad session, it was postponed on the stated ground that comparison against a high-variance architecture would "waste its 3.5 h". Every later plan re-gated it behind a full capability-gate pass — the ladders of Experiments 8, 9, and 10 repeat the condition verbatim — and the final dense-supervision plan demoted it to an abort-branch contingency, where it was in any case absent from the launched notebook. The condition was self-sealing: the battery required the architecture to work before it would authorize the test of whether the architecture does anything. Ten experiments, zero full-battery passes, zero authorizations.

The shipped implementation of the control is defective in three ways. It is budget-reduced, not parameter-matched: at rank 16 it trains roughly 10.1M parameters against Experiment 10's 132,227,208, a ∼13× shortfall; matching needs rank ≈210, which is a design change, not a flag. Its freeze predicate leaks: parameters under the backbone layers are spared, so the family-routed experts and the per-layer prefix gates keep training, and the shipped "plain LoRA" is not plain. And it addresses only the capability question; the selective-prediction control — plain LoRA with its own mean-log-probability abstention at matched coverage, the decisive baseline for the thread of Section 5 — was never specified anywhere in the program. There are two missing controls, not one.

The remaining limitations are stated briefly.

- **Two seeds.** Every result rests on seeds 17 and 29 (n=2); the planned replication seeds 41 and 73 never ran.
- **Twenty days.** The dated record spans 2026-08-17 to 2026-09-05, about 20 GPU sessions and 55–65 dual-T4 session-hours (Appendix B); the explored region of designs, budgets, and hyperparameters is correspondingly narrow.
- **Easy audit pools.** 58–59% of audit rows were trivially easy, so any monotone confidence score yields near-perfect selective precision at the observed coverages; the flawless operating points carry little evidential weight.
- **Small samples.** Instrument validation rests on n=8 rows per grader family and n=8 intervention records.
- **One researcher, one codebase.** The model, gates, graders, and analyses share an author and a repository; Section 7 documents what that risk cost in practice, and no independent reimplementation exists.
- **Budget undecidability.** The terminated dense-supervision run reached 125 optimizer updates on 2,000 row-presentations, against 2×10^5 to 1.5×10^7 in published successes of its class; that question closes undecidable, not falsified.

9. What We Would Do Differently

Run the novelty audit before building. The audit that ended this program took about forty minutes: twelve search agents, roughly 200 queries, adversarial refutation of each claim. It scored six architectural claims against prior art and returned zero survivors at a mean of 0.63 of 5 (Section 6). The program it terminated took twenty days, about 20 GPU sessions, and 55–65 dual-T4 session-hours. The verdict depended on nothing the experiments produced; the same searches, run before the first GPU hour, would have returned the same result. Forty minutes of search against twenty days of experiments is not a close call. The literature search is the cheapest experiment available and the only one that cannot return an uninterpretable null; it goes first.

Certify instruments before trusting measurements. The program executed roughly 284 preregistered gate evaluations through instruments carrying about 30 documented defects in five classes (Section 7), and the sharpest surfaced only in the final two days: the expert graph was called zero times by the trainer, routing was a gold-label lookup, two of fourteen gates in the production caller were hardcoded to pass, and the one preregistered abort that fired was retracted as vacuous because its selector read a key the pipeline never emits. Preregistration disciplined what we asked; it did nothing to establish that the code measured

the system we described. A certification battery — liveness, sign, and sensitivity checks on every grader, gate, and ablation — must run and pass before the first preregistered gate is trusted, and its passing is itself an artifact to version and report.

Never gate a control on the success of the system it controls for. Section 8 records the drift: unconditional, then postponed, then gated behind a full battery pass, then an abort-branch contingency — and ten experiments with zero full passes therefore authorized a four-hour experiment zero times. The structure of the error generalizes: a control is most informative exactly when the system fails its gates, so conditioning the control on gate success guarantees it runs only when it is least needed and never when it is decisive. Controls run unconditionally, at fixed budget, scheduled before the comparisons that depend on them; an authorization ladder may gate scale-ups and benchmarks, never controls.

Run the power analysis before choosing audit-set sizes. At 288 audit rows per seed, the headline abstention comparison could only have detected an effect roughly three times the size observed — a net six rows in 300 published answers. The original bootstrap bar of 0.90 was, by the program's own later plan, set with no power analysis; the first pre-computed achievable bar arrived only with the tenth experiment's plan; and the selection-arm comparison was voided outright with 6 and 4 discordant pairs against a preregistered floor of 25. The minimum detectable effect of each of these designs was computable from the audit-set size before Experiment 1. Any successor to the selective-prediction thread inherits an audit-set requirement in the thousands of rows, not hundreds, and should derive that number before committing to a design.

10. Conclusion

This program set out to build a latent workspace and instead measured, with progressively better instruments, that the workspace could be read but not used. Three things in the record are established well enough to hand on. First, a constructed read–write dissociation that survived repair: with the premise physically deleted from the decoder's input, masked generation accuracy was exactly 0.000 on 99 of 99 rows on both seeds — zero leaked rows in 198 masked audits — while a linear probe recovered 0.297 and 0.334 of the withheld premise, below its own preregistered 0.50 bar. Lagged Coupling [1] reports that internal representations become readable before they become causal in pretrained models; this program extends that observation to a deliberate, instrumented attempt to close the gap, across ten experiments and every repair the record demanded, and the gap did not close. Second, a prior-art map (Section 6) on which three narrow cells remain literally unoccupied (Section 6.1), carried with its own caveat: an unoccupied combination is a contribution only if occupying it produces an effect, and this record measured none. Third, an instrument-failure taxonomy with detectors (Section 7) that transfers to any small training program at zero GPU cost.

The record establishes no capability claim and no superiority claim. The matched control never ran, so nothing here separates the architecture from finetuning at all; the one thread that kept passing its operational gates cleared its own significance bar once in six seed-runs (a two-row pass on eleven published rows); across Experiments 9–10 the net difference was six rows in 300 published answers, and the claim that it beats the free baseline is not supported, and we do not advance it; the dense-supervision question closes undecidable at 125 optimizer updates, not falsified.

What remains is an argument about genre. A falsification record with intact instruments is cheap for the field to reuse and expensive for any one researcher to produce: the gates, the graders, the defect detectors, and the map of already-occupied territory were paid for once, over twenty days and roughly 284 gate evaluations, and all of them transfer at no cost to whoever next attempts this cell of the design space. This record is offered as such.

References

[1] X. Xun. Lagged coupling: Internal representations become readable before they become causal. arXiv:2609.01048, September 2026.

[2] X. Xun. Evidence-type competition: When can interventional data teach language models causal direction? arXiv:2607.29484, 2026.

[3] X. Xun. Causal structure is inducible but functionally decoupled: The routing/readout boundary of a typed mechanism library. arXiv:2608.11767, 2026.

[4] X. Xun. Located but not releasable: Silent gate inversion and bounded linear release. arXiv:2608.11822, 2026.

[5] J. Liu et al. Dual-cache latent space communication between heterogeneous language models. arXiv:2608.20617, 2026.

[6] Z. Shen, H. Yan, L. Zhang, Z. Hu, Y. Du, and Y. He. CODI: Compressing chain-of-thought into continuous space via self-distillation. *Empirical Methods in Natural Language Processing*, 2025. arXiv:2502.21074.

[7] S. Hao, S. Sukhbaatar, D. Su, X. Li, Z. Hu, J. Weston, and Y. Tian. Training large language models to reason in a continuous latent space. *Conference on Language Modeling*, 2025. arXiv:2412.06769.

[8] M. I. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6(2):181–214, 1994.

[9] N. Shazeer et al. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *International Conference on Learning Representations*, 2017. arXiv:1701.06538.

[10] C. Rosenbaum, T. Klinger, and M. Riemer. Routing networks: Adaptive selection of non-linear functions for multi-task learning. *International Conference on Learning Representations*, 2018. arXiv:1711.01239.

[11] H. Tang, J. Liu, M. Zhao, and X. Gong. Progressive layered extraction (PLE): A novel multi-task learning (MTL) model for personalized recommendations. *ACM Conference on Recommender Systems*, 2020.

[12] W. Fedus, B. Zoph, and N. Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022. arXiv:2101.03961.

[13] S. Rajbhandari et al. DeepSpeed-MoE: Advancing mixture-of-experts inference and training to power next-generation AI scale. *International Conference on Machine Learning*, 2022. arXiv:2201.05596.

[14] A. Chronopoulou, M. E. Peters, and J. Dodge. Efficient hierarchical domain adaptation for pretrained language models. *North American Chapter of the Association for Computational Linguistics*, 2022. arXiv:2112.08786.

[15] D. Dai et al. DeepSeekMoE: Towards ultimate expert specialization in mixture-of-experts language models. arXiv:2401.06066, 2024.

[16] X. Wang et al. HoME: Hierarchy of multi-gate experts for multi-task learning at Kuaishou. arXiv:2408.05430, 2024.

[17] J. Han et al. MoRAgent: Parameter efficient agent tuning with mixture-of-roles. arXiv:2512.21708, 2025.

[18] W. Lee et al. MapCoder-Lite: Distilling multi-agent coding into a single small LLM. arXiv:2509.17489, 2025.

[19] S. Yang et al. Phase-aware mixture of experts for agentic reinforcement learning. arXiv:2602.17038, 2026.

[20] B. Taetz, F. da Silva, and G. Bleser-Taetz. Towards continual motion-language agents: LoRA variants for incremental motion understanding and generation. arXiv:2606.30266, 2026.

[21] Kimi Team. Kimi K2: Open agentic intelligence. arXiv:2507.20534, 2025.

[22] Moonshot AI. Kimi K2.5 repository. https://github.com/MoonshotAI/Kimi-K2.5, 2026. Accessed 2026-09-05.

[23] Moonshot AI. Kimi-K2.6 model card. https://huggingface.co/moonshotai/Kimi-K2.6, 2026. Accessed 2026-09-05.

[24] MarkTechPost. Moonshot AI releases Kimi K2.6 with long-horizon coding, agent swarm scaling to 300 sub-agents and 4,000 coordinated steps. https://www.marktechpost.com/2026/04/20/moonshot-ai-releases-kimi-k2-6-with-long-horizon-coding-agent-swarm-scaling-to-300-sub-agents-and-4000-coordinated-steps/, April 2026. Accessed 2026-09-05.

[25] Pandaily. Deconstructing the Kimi K3 technical report: Moonshot AI starts setting problems for DeepSeek as two scaling axes redefine the frontier. https://pandaily.com/kimi-k3-technical-report-analysis-deepseek-jul2026, July 2026. Accessed 2026-09-05.

[26] J. Koutník, K. Greff, F. Gomez, and J. Schmidhuber. A clockwork RNN. *International Conference on Machine Learning*, 2014. arXiv:1402.3511.

[27] J. Chung, S. Ahn, and Y. Bengio. Hierarchical multiscale recurrent neural networks. *International Conference on Learning Representations*, 2017. arXiv:1609.01704.

[28] M. Chen et al. Parallel scaling law for language models. arXiv:2505.10475, 2025.

[29] Z. Wang, X. Dang, R.-J. Zhu, et al. Learning what to remember: Test-time training via context distillation. arXiv:2608.01672, 2026.

[30] S. Eyuboglu et al. Cartridges: Lightweight and general-purpose long context representations via self-study. arXiv:2506.06266, 2025.

[31] A. Graves. Adaptive computation time for recurrent neural networks. arXiv:1603.08983, 2016.

[32] M. Dehghani, S. Gouws, O. Vinyals, J. Uszkoreit, and Ł. Kaiser. Universal transformers. *International Conference on Learning Representations*, 2019. arXiv:1807.03819.

[33] J. Austin et al. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 2021. arXiv:2107.03006.

[34] S. Nie et al. Large language diffusion models. arXiv:2502.09992, 2025.

[35] M. Arriola et al. Block diffusion: Interpolating between autoregressive and diffusion language models. arXiv:2503.09573, 2025.

[36] DiffusionGemma Team et al. DiffusionGemma technical report. arXiv:2608.00146, 2026.

[37] V. Vovk, A. Gammerman, and G. Shafer. *Algorithmic Learning in a Random World*. Springer, 2005.

[38] Y. Geifman and R. El-Yaniv. Selective classification for deep neural networks. *Advances in Neural Information Processing Systems*, 2017. arXiv:1705.08500.

[39] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks. *International Conference on Machine Learning*, 2017. arXiv:1706.04599.

[40] A. N. Angelopoulos and S. Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. arXiv:2107.07511, 2021.

[41] L. Feng et al. Towards better selective classification. *International Conference on Learning Representations*, 2023. arXiv:2206.09034.

[42] I. Galil, M. Dabbah, and R. El-Yaniv. What can we learn from the selective prediction and uncertainty estimation performance of 523 ImageNet classifiers? *International Conference on Learning Representations*, 2023. arXiv:2302.11874.

[43] J. Traub et al. Overcoming common flaws in the evaluation of selective classification systems. *Advances in Neural Information Processing Systems*, 2024. arXiv:2407.01032.

[44] A. A. Phalod. When should a language model trust itself? Same-model self-verification as a conditional confidence signal. arXiv:2605.02915, 2026.

[45] X. Wang et al. Self-consistency improves chain of thought reasoning in language models. *International Conference on Learning Representations*, 2023. arXiv:2203.11171.

[46] E. van Miltenburg, C. van der Lee, and E. Krahmer. Preregistering NLP research. *North American Chapter of the Association for Computational Linguistics*, 2021. arXiv:2103.06944.

[47] F. Tambon et al. Silent bugs in deep learning frameworks: An empirical study of Keras and TensorFlow. arXiv:2112.13314, 2021.

A. Per-Experiment Gate Tables

Experiments 1–5 gate detail

Numbers are as recorded in the per-experiment evaluation artifacts. Experiments 2–5 are reproducible from shipped JSON; the structural diagnostic (no gate battery; seed 7; 0/320 sequences committed) and Experiment 1 have no on-disk results record. Preregistered gate failures are marked red; green marks the record's only routing-gate passes.

Table 9: Experiment 1 (substrate transplant): key outcomes. Behavioural rows are from the better seed (17); seed 29's behavioural numbers were never recorded. No named sub-gate battery existed yet; both capability gates failed.

Measurement	Observed	Note
Training completion (per seed)	896 microsteps, 0 skipped	both seeds
Prompt leakage / quality pass	0 / 0.75	one quarter of checks failed
Narrow content accuracy	0.50	exact-phrase false negatives
Candidate vs. frozen-base token F1	0.469 vs. 0.312	no superiority claim supported
Commitment coverage	0	every answer withheld
Mean lane-summary cosine	0.947	lanes nearly collapsed
Capability gates	failed	checkpoint lost (session restart)

Table 10: Experiment 2 (graded evaluation): per-seed gate detail. Audit: 32 rows per seed (16 behaviour anchors, 16 corpus records).

Measurement	Seed 17	Seed 29	Gate
Completed microsteps / skipped	1,024 / 0	1,024 / 0	exact / 0
Quality pass / prompt leakage	0.750 / 0	0.781 / 0	—
Probe semantic accuracy	0.625	0.563	≥ 0.75
Safe-abstention accuracy	0	1.000	seed-sensitive
Probe commitment accuracy	0.625	0.313	—
Commitment coverage	0.125	0.125	4 of 32 published
Teacher-forced balanced accuracy (s29)	0.918		mismatch vs. 0.125 coverage
Mean lane-summary cosine	0.496	0.331	down from 0.947
Complete capability gate	failed	failed	—

Table 11: Experiment 3 (publication boundary): per-seed gate detail. Audit: 64 outputs per seed (32 anchors, 32 corpus records).

Measurement	Seed 17	Seed 29	Gate
Completed microsteps / skipped	4,224 / 0	4,224 / 0	exact / 0
Quality pass / prompt leakage	0.813 / 0	0.891 / 0	—
Probe semantic accuracy	0.625	0.813	≥ 0.75
Safe-abstention accuracy / committed rate	1.000 / 1.000	0.875 / 0.875	—
Probe commitment accuracy	0.625	0.750	≥ 0.70
Mean lane-summary cosine	0.707	0.805	< 0.90
Generated-emission calibration (neg. reject)	failed	0.583	≥ 0.70
Fitted thresholds (s29)	commit 0; risk/verifier $\approx 10^{-3}$		degenerate
One-expert routing share	98.4%	two experts material	gate too weak
Complete capability gate	failed	failed	—

Table 12: Experiment 4 (on-policy calibration): per-seed gate detail, 18-sub-gate battery. Both seeds passed 15 of 18 and failed the same three routing gates. Intervention: $n = 8$ records per seed, confounded with head saturation.

Measurement	Seed 17	Seed 29	Gate
Completed microsteps / skipped	4,224 / 0	4,224 / 0	exact / 0
Prompt leakage	0.0	0.0	$= 0$
Quality pass rate	0.938	0.906	≥ 0.75
Probe semantic accuracy	0.875	0.844	≥ 0.75
Safe-abstention accuracy / committed rate	1.00 / 1.00	1.00 / 1.00	≥ 0.75 / 0.70
Probe commitment accuracy	0.844	0.906	≥ 0.70
Candidate token F1 (frozen base 0.273)	0.571	0.505	$\geq 0.8\times$ base
Mean lane-summary cosine	0.860	0.831	< 0.90
Calibration: positive accept	1.000	0.982	≥ 0.70
Calibration: negative reject (Exp. 3: 0.583)	0.889	0.917	≥ 0.70
Calibration: balanced accuracy	0.944	0.949	≥ 0.70
Second-route load	0.063	0.094	≥ 0.10
Normalized route entropy	0.130	0.174	≥ 0.25
Routing intervention $\lvert\Delta\text{commit}\rvert$	0.0035	0.0008	≥ 0.01
Fitted commitment threshold	0.0	0.0	degenerate (2nd recurrence)
Complete capability gate	failed (3 of 18)	failed (3 of 18)	—

Table 13: Experiment 5 (routing redesign): per-seed gate detail, same 18-sub-gate battery (gate keys byte-identical to Experiment 4). The seeds fail differently: seed 17 fails 8 of 18, seed 29 fails 2 of 18.

Measurement	Seed 17	Seed 29	Gate
Completed microsteps / skipped	4,224 / 0	4,224 / 0	exact / 0
Prompt leakage	0.266	0.0	$= 0$
Quality pass rate	0.578	0.844	≥ 0.75
Probe semantic accuracy	0.281	0.813	≥ 0.75
Safe-abstention accuracy / committed rate	0.75 / 0.75	1.00 / 1.00	≥ 0.75 / 0.70
Probe commitment accuracy	0.656	0.625	≥ 0.70
Candidate token F1 (frozen base 0.307)	0.324	0.585	$\geq 0.8\times$ base
Causal-path token F1 (diagnostic)	0.588	0.589	n/a
Mean lane-summary cosine	0.942	0.890	< 0.90
Calibration: accept / reject / balanced	0.818 / 1.000 / 0.909	0.875 / 0.933 / 0.904	≥ 0.70
Second-route load	0.000	0.418	≥ 0.10
Normalized route entropy	0.000	0.570	≥ 0.25
Routing intervention $\lvert\Delta\text{commit}\rvert$	0.0035	0.0073	≥ 0.01
Fitted commitment threshold	0.912 (non-degen.)	0.0	degenerate (3rd recurrence)
Complete capability gate	failed (8 of 18)	failed (2 of 18)	—

Table 14: Experiment 6 gate detail (Experiment 5 values in parentheses). Telemetry defect: the uploaded run telemetry resumed past the training window, so whether the diversity penalty engaged under nonzero init is unmeasured, not answered.

Gate	Seed 17	Seed 29
Completed microsteps / skipped (exact / 0)	4,224 / 0	4,224 / 0
Prompt leakage ($= 0$)	0.0 (was 0.266)	0.0
Probe semantic accuracy (≥ 0.75)	0.781 (was 0.281)	0.750
Safe-abstention accuracy (≥ 0.75)	1.00	0.875
Committed rate (≥ 0.70)	1.00	0.875
Probe commit accuracy (≥ 0.70)	0.781 (was 0.656)	0.625 (was 0.625)
Lane summary cosine (< 0.90)	0.902 (was 0.942)	0.919 (was 0.890)
Second-route load (≥ 0.10)	0.000	0.113 (was 0.418)
Normalized route entropy (≥ 0.25)	0.000	0.306 (was 0.570)
Routing intervention $\lvert\Delta\,\text{commit}\rvert$ (≥ 0.01)	0.0012	0.0031
Fitted commitment threshold (non-degenerate, diagnostic)	0.0	0.090
Complete capability gate	failed (4 of 18)	failed (4 of 18)

Table 15: Experiment 7 gate detail (Experiment 6 values in parentheses). Complete telemetry showed the diversity penalty engages under nonzero init but saturates below the weight's bite point; the anchor-stratified canary was blind by construction to route diversity.

Gate	Seed 17	Seed 29
Completed microsteps / skipped (exact / 0)	4,224 / 0	4,224 / 0
Prompt leakage (= 0)	0.0	0.0625 (was 0.0)
Probe semantic accuracy (≥ 0.75)	0.563 (was 0.781)	0.469 (was 0.750)
Probe commit accuracy (≥ 0.70)	0.844 (was 0.781)	0.469 (was 0.625)
Lane summary cosine (< 0.90)	0.928 (was 0.902)	0.829 (was 0.919)
Second-route load (≥ 0.10)	0.000	0.273 (was 0.113)
Normalized route entropy (≥ 0.25)	0.000	0.846 (was 0.306)
Routing intervention $\lvert\Delta$ commit$\rvert$ (≥ 0.01)	0.0082 (was 0.0012)	0.0037
Head-phase valid positives (floor 16/family)	78 of 512 attempts	77 of 448 attempts
Fitted commitment threshold (non-degenerate)	0.0	0.0
Complete capability gate	failed (6 of 18)	failed (5 of 18)

Table 16: Experiment 8 gate detail. Seed 29's safe-abstention 0.000 is over-conservatism, not miscalibration: corrupt rejection rate 0.922, commitment on 7.8% of audit rows and none of the commit-expected probes. Seed 29's verifier head ranked corrupt candidates below clean on audit (margin −0.109) despite correct ranking at policy-fit time.

Gate	Seed 17	Seed 29
Validation calibration balanced accuracy (≥ 0.70)	0.884	0.877
Fitted publish threshold (non-degenerate)	0.578	0.689
Turn-scaffold leak rate (= 0)	0.188	0.016
Quality pass rate (≥ 0.75)	0.563	0.688
Probe content accuracy (≥ 0.75)	0.250	0.500
Probe commit accuracy (≥ 0.70)	0.656	0.406
Safe-abstention accuracy (≥ 0.75)	0.625	0.000
Lane summary cosine (< 0.90)	0.945	0.968
Workspace / causal token F1 (report)	0.289 / 0.560	0.454 / 0.570
Read-out ablation Δ graded accuracy (≥ 0.05)	0.167	0.306
Fan-in vs own greedy (≥ 0)	+0.222	+0.028
Fan-in vs self-consistency (headline)	0.455 vs 0.758	0.515 vs 0.818
Selective accuracy, heads vs log-prob (matched coverage)	0.818 vs 0.636 (cov. 0.306)	0.333 vs 1.000 (cov. 0.083)
Complete capability gate	failed (9 of 20)	failed (11 of 20)

Table 17: Experiment 9 gate detail (24 gates; 320 audit rows per seed: 256 behavior anchors plus 64 text-to-SQL). Preregistered gate failures in red. The prefix-gate row is telemetry disqualified as evidence (RMSNorm scale invariance). The validity-floor row records the vacuous pass: the shipped gate checked that head training occurred, not that the floor of 16 valid positives per family was met.

Measurement (gate)	Seed 17	Seed 29
Microsteps completed / skipped	4,224 / 0	4,224 / 0
Turn-scaffold leak rate (gate = 0)	0 / 320	0 / 320
Causal − workspace greedy accuracy (parity within 0.05)	+0.013	−0.003
Workspace / causal token-F1 ratio (gate ≥ 0.9)	1.015	1.025
Memory-token ablation Δ graded accuracy (gate ≥ 0.05)	0.000	−0.022
Memory-token ablation Δ token F1	0.0003	−0.003
Prefix gate tanh(α) full-run range (telemetry)	0.0500–0.0507	0.0498–0.0504
Medium-bin exactly-one-correct fraction (gate ≥ 0.10)	0.000	0.000
Weighted vs. plurality SC discordant pairs (pooled mid-p 0.125)	0 / 0	2 / 0
Conformal coverage (band $[0.20, 0.50]$)	0.244	0.188
Published answers correct	78 / 78	60 / 60
Selective accuracy at matched coverage, heads vs. log-prob	1.000 vs. 0.987	1.000 vs. 0.983
AUGRC, heads vs. log-prob (heads must be lower)	0.0637 vs. 0.0594	0.0654 vs. 0.0623
Paired-bootstrap win fraction (bar 0.90)	0.13	0.179
Spearman(publish score, mean log-prob) (gate < 0.95)	0.830	0.821
Selective-risk UCB_{95} (gate ≤ 0.45)	0.038	0.049
Safe-abstention probes	1.000	1.000
Text-to-SQL commit coverage (quality 0.078 / 0.109)	0.000	0.000
Pooled quality (gate ≥ 0.70)	0.663	0.666
Lane summary cosine (gate < 0.90; failures eight and nine)	0.940	0.975
On-policy valid positives, numeric/ordering/unit (floor 16 each)	2 / 5 / 2	1 / 2 / 5
Gates passed (of 24)	17	17

Table 18: Experiment 10 gate detail (22 gates; 288 audit rows per seed). The first launch crashed at the same calibration line on both seeds (a feature-count guard retained from Experiment 9 against the preregistered four- and five-feature rule forms); after a two-line fix the resume was bit-equivalent to an uninterrupted run, with the 41 and 43 valid emissions of 512 reproduced digit for digit. The canary-cleanliness gate read the resumed process's log (containing only the resume marker) and recorded false on both seeds; recomputing its exact rule on the original training logs gives seven canary records per seed, zero dirty boundaries, and zero prompt leaks, so it is excluded from the tally as an artifact.

Measurement (gate)	Seed 17	Seed 29
Microsteps completed / skipped	3,200 / 0	3,200 / 0
Masked-row premise leaks, 99 rows (gate $= 0$)	0	0
Causal floor, prefix removed, masked rows	0.000	0.000
Masked workspace accuracy (terminal-rung gate > 0)	0.000	0.000
Trained gist attribution control, masked rows	0.000	0.000
Latent-only premise probe (gate ≥ 0.50)	0.297	0.334
Unmasked parity gap, workspace − causal	+0.037	0.000
Verifier margin on generated candidates (gate > 0)	−0.277	−0.227
On-policy validity floor (gate ≥ 16 per family)	41 / 512	43 / 512
Conformal coverage (band $[0.20, 0.55]$)	0.295	0.267
Publish-rule distinct scores (non-degeneracy)	329	372
Published answers correct	83 / 85	77 / 77
Selective-risk UCB_{95} (gate ≤ 0.15)	0.072	0.038
Safe-abstention accuracy / commit rate	1.000 / 1.000	1.000 / 1.000
Partial AUGRC, heads vs. log-prob (band $[0.05, 0.50]$)	0.0128 vs. 0.0156	0.0073 vs. 0.0114
Full-range AUGRC, heads vs. log-prob	0.12731 vs. 0.14122	0.13871 vs. 0.12646
Paired-bootstrap win fraction, partial (bar 0.80)	0.746	0.792
Paired-bootstrap win fraction, full range	0.994	0.039
Canary-cleanliness gate (artifact; recomputed clean)	7 records, 0 dirty	7 records, 0 dirty
Gates passed (of 22; canary artifact excluded)	15	15

B. Timeline and Artifact Provenance

The main-text timeline is Table 2. This appendix records the mapping between the paper's experiment numbering and the program's internal artifact tags, and the provenance status of each experiment's numbers.

Table 19: Experiment numbering, internal artifact tags, and provenance status. "Shipped JSON" means the per-seed `checkpoint-evaluation.json` and capability-gate files exist on disk and reproduce every number reported here; "no on-disk record" means the numbers survive only in session logs and are flagged wherever used.

Experiment	Artifact tag	Provenance
Structural diagnostic	`legacy-cpu`	No on-disk record; plumbing check only.
1	`v5.2`	Checkpoint lost to a session restart; build report on disk; behavioural numbers from logs only (best seed).
2	`v5.3`	Shipped JSON, both seeds.
3	`v5.4`	Shipped JSON, both seeds.
4	`v5.5`	Shipped JSON, both seeds; 18 named sub-gates; values verified to the recorded decimal.
5	`v5.6`	Shipped JSON, both seeds; gate keys byte-identical to Experiment 4's.
6	`v5.6.1`	Shipped JSON, both seeds; run telemetry incomplete (resumed past the training window).
7	`v5.6.2`	Shipped JSON, both seeds.
8	`v6.0`	Shipped JSON, both seeds; one session crash recovered byte-identically (all six policy thresholds matched to full float precision).
9	`v8.0`	Shipped JSON, both seeds; two post-session instrument corrections disclosed in the text.
10	`v9.0`	Shipped JSON, both seeds; first launch crashed at calibration on both seeds, resume bit-equivalent (on-policy counts reproduced digit for digit); one canary gate certified out-of-band.
—	`v10.0`	Dense-supervision successor; launched 2026-09-05 and terminated at ∼125 optimizer updates after the novelty audit; no audit battery was ever evaluated. Closes undecidable.

Two experiments therefore have numbers that cannot be independently regenerated: the structural diagnostic and Experiment 1. Both are used in this paper only as narrative (what the program believed at the time and what it changed next), never as evidence for any surviving claim. All numbers for Experiments 2–10 trace to shipped, SHA-256-pinned artifacts evaluated by fresh-process reload.

The substrate was pinned to `Qwen/Qwen3-0.6B-Base` throughout. Seeds were 17 and 29 in every GPU experiment; the planned replication seeds 41 and 73 never ran. The program's corpus grew from 4,300 unadjudicated episodes (Experiment 1) to 7,498/945/841 train/validation/test rows with 1,699 execution-verified benchmark episodes (Experiment 5 onward); official evaluation splits of external benchmarks were never trained on.