WHEN ROUTING MUST BE RIGHT: IRREVERSIBLE LOCK-IN IN ATTENTION ROUTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Hybrid sequence models pair a cheap recurrent path with a small budget of attention, and something has to decide which tokens get it. On multi-query associative recall that decision is worth three orders of magnitude: spending a fixed 6.25% budget on the right tokens gives 0.997 accuracy, a static schedule tuned to the same budget gives 0.0002. Learned routers are notoriously hard to train, and the usual explanation, that top-k selection is discontinuous, so unselected candidates get no gradient, is incomplete. REINFORCE supplies unbiased gradient to unselected tokens and still lands at chance, and one router that had been correct and stable for $25\,000$ steps produced no improvement at all. We argue the variable that matters is timing, and measure it. Holding the number of correctly-routed steps fixed and varying only their position, misrouting the first 8000 steps leaves accuracy at chance while delaying the identical window by fifty steps recovers almost everything; the failing arm is the one with more time to recover afterwards. Ten seeds at each of nine onsets put the half-way point at 23 steps, and a sweep over the deficit's *length* puts its threshold at 48, two measurements agreeing on the scale from opposite directions, and settling the outcome inside the first 0.06% of training. The obstruction is that the signal a router needs does not exist when it is needed: the marginal utility of attention is 0.0000 ± 0.0000 under misrouting and 2.72 under correct routing. We probe four ways for a usable substitute and none survives, because each returns the model's existing reliance on attention, which is itself a product of the routing it was trained under. Two explanations that would make this an artifact are ruled out by measurement: learned routing stays at chance with four times the budget, and a four-block model in which attention feeds later positions locks in more readily than one block, not less. We also give a mechanism, and a pre-registered prediction it makes about learning-rate scaling that we test and falsify.

1 INTRODUCTION

Conditional computation has to choose what gets computed, and a router makes that choice. How much rides on it is easy to underestimate. In the setting we study, two policies that spend an identical attention budget differ in accuracy by three orders of magnitude.

The difficulty of training such routers is well documented. Rosenbaum et al. (2019) put it plainly: at initialisation "the router needs to learn the value of the modules, while these are effectively randomly trained and have not yet taken on any real meaning," and the resulting "chicken-and-egg" problem "can result in the routing dynamics never stabilizing, leaving the network at effectively random performance." The same paper suggests a remedy, "fixing their parameters for a short initial period, or by reducing their learning rate."

The prevailing modern explanation is narrower and mechanical. Because top-k selection is discontinuous, a candidate that goes unselected receives no gradient pushing it toward selection. This has motivated differentiable relaxations, and it shapes the design of Mixture-of-Depths (Raposo et al., 2024), which avoids the issue by computing top-k non-causally across the whole sequence during training and then distilling a causal predictor from those selections.

We think that explanation is incomplete in both directions. Restoring gradient flow turns out to be neither necessary nor sufficient. Section 3 reports a REINFORCE router that is unbiased and does

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

reach unselected tokens, and that nonetheless sits at chance after $40\,000$ steps. Section 5.1 describes the observation that prompted the rest of this work: a run whose router was correct and stable for $25\,000$ steps, and which still scored 0.0000. On a diagnosis that attributes failure to unstable routing, that run should have succeeded.

Our argument is that the variable which matters is timing. Once the backbone has settled on a solution that ignores attention, routing that later becomes correct arrives too late to be of use. We make this precise using the deficit-window protocol of Achille et al. (2019) (Section 6), while noting there that the sensitivity profile we measure is not the non-monotonic one that characterises critical periods in their setting.

Production MoE systems appear to sit in the regime this describes. Routing saturates within the first 1% of pretraining, before the experts it selects between have matured (Xue et al., 2024; Muennighoff et al., 2024), and since neither of those studies manipulates timing, neither can say what the resulting lock-in costs. We ran no MoE experiments of our own: the parallel is what led us to the question, not evidence about the answer. Outside the synthetic task, what we do show is the necessity result of Section 4, which holds on real text with dense supervision.

What is new, and how firmly we hold it. Neither the difficulty of training routers nor the staged fix is ours; Rosenbaum et al. (2019) has both. What is new is the measurement. *Established*: the four methods of Section 3 fail; the quantity each implicitly estimates, δ_t, is identically zero exactly when it is needed and abundant only once routing is already correct, on the synthetic task and on real text with dense supervision; misrouting confined to the first $\sim$48 steps costs most runs the task while delaying the identical window past $\sim$50 steps costs nothing; a learned router's attention pathway follows the same suppression trajectory as an imposed deficit; and no probe we construct recovers the missing signal, because each reports the routing the model has already been given. Two explanations that would make this an artifact are excluded by measurement: learned routing stays at chance with four times the budget, and a four-block model in which attention feeds later positions locks in more readily than one block. *Suggested*: the mechanism of Section 7, and the parallel to mixture-of-experts lock-in. *Not established*: that no routing method can succeed — we tested a family, not the space — that any of this holds at greater scale or on natural data, or the prediction of Section I, which we falsify ourselves. The effect also needs a sufficiently high learning rate: a deficit fatal at 3×10^{-3} leaves every run at ceiling at 1×10^{-3}. The upshot is a constraint on method design rather than a method: external routing information is necessary, and must be near-perfect and permanent.

2 SETUP

Task. We use multi-query associative recall (MQAR), ported from the Zoology benchmark (Arora et al., 2023) without algorithmic modification, with a vocabulary of 8192, sequences of length 256 and 16 key–value pairs. Before using the harness for any new claim we checked that it reproduces published behaviour: attention reaches 0.9998, and Mamba shows the capacity wall reported in that work.

Model. One block: embedding $\to$ causal depthwise convolution $\to$ [Mamba (Gu & Dao, 2023) $\|$ causal attention] $\to$ FFN $\to$ head. Attention is added to the SSM output rather than substituted for it. That detail turns out to matter more than one might expect, and Section P shows it alone can move measured accuracy from 0.008 to 0.983.

Budget. Every policy selects exactly 16 of 256 positions ($\rho = 0.0625$). "q-recall" is the fraction of true query positions selected; chance is 0.0625. Accuracy chance is $1/8192 = 1.22 \times 10^{-4}$.

Regime. Headline results are at $d_{\text{model}} = 128$, a setting in which the task is strongly attention-limited. Holding the mask fixed from step 1, an SSM-only model reaches 0.0001 where oracle routing reaches 0.9983.

Table 1: All arms at $40\,000$ steps, the budget at which the oracle arm converges, reported as mean $\pm$ s.d. over 3 seeds. Every learned method sits at or barely above chance selectivity (accuracy chance is 1.22×10^{-4}; q-recall chance is 0.0625). No arm was bimodal across seeds, so means are reported; where a result *is* bimodal elsewhere in this paper we give individual runs instead.

Method	Accuracy	q-recall
SSM only (no attention)	0.0001 ± 0.0001	—
Static schedule (tuned to canonical positions)	0.0003 ± 0.0003	0.144
Straight-through estimator	0.0003 ± 0.0001	0.059
Marginal-utility auxiliary regressor	0.0004 ± 0.0001	0.023
MoD-style continuous gating	0.0001 ± 0.0001	0.000
REINFORCE + EMA baseline + entropy bonus	0.0002 ± 0.0002	0.069
Oracle	$\mathbf{0.9962 \pm 0.0024}$	**1.000**

3 Four standard router-training methods fail

Table 1 reports the four standard approaches. Each fails for a separately verified reason rather than from insufficient tuning.

Straight-through approximates a step function as locally smooth; the true landscape is flat almost everywhere and spikes only at correct tokens. The marginal-utility router regresses loss(no attn) $-$ loss(dense attn) per token; instrumenting the target shows it stays at ≈ 0.000 at both query and filler positions throughout, because attention is never exercised on the tokens that matter. MoD-style gating gives real gradient to a selected token's magnitude, but selection runs under `no_grad`, so an unselected token receives zero gradient toward ever being selected.

REINFORCE is the row that carries the argument. It is unbiased and does reach unselected tokens, so its failure is what rules out the pure gradient-flow account. Its task loss stays pinned at 8.32 ± 0.01 across all $40\,000$ steps: variance from an 8192-way cross-entropy reward swamps the signal entirely.

Why Mixture-of-Depths does not encounter this. Raposo et al. (2024) train the deployable router on labels derived from a non-causal top-k, so their training-time router is approximately correct by construction and never enters the deadlock. Their own explanation for why that works is a direct statement that *supervised* routing is easy while *task-loss* routing is hard — the separation Section 5 isolates, and Appendix C quotes.

4 The routing signal does not exist during the window

Having seen four standard methods fail, the natural question is whether some better optimiser-side intervention would succeed. We think it would not, and the reason can be measured rather than argued. During the critical window, the information a router would need is not merely difficult to extract; it is not there.

What a router would need. What tells a router which positions to select is the marginal utility of attention at each one, $\delta_t = \ell_t(\text{no attention}) - \ell_t(\text{dense attention})$. Each of the four methods in Section 3 is, in effect, estimating the sign of δ_t — the straight-through estimator and MoD gating through the gradient that reaches the gate, REINFORCE through the sampled return, the marginal-utility regressor by fitting δ_t explicitly. We do not claim this exhausts the ways a router could be trained; it characterises the family we tested, which is the family in common use. So we measured δ_t directly, under two training regimes that differ only in the routing mask.

The deadlock, as a number. Rosenbaum et al. (2019) describe the chicken-and-egg problem qualitatively: the router cannot learn the value of modules that are themselves untrained. Figure 1 puts numbers on it (Table 3 lists them). Under misrouting, δ is 0.0000 ± 0.0000 at step 4000, zero to four decimal places with vanishing variance across seeds, while under correct routing it reaches 2.72 ± 0.51. The signal a router would need in order to learn only becomes available once routing is already correct.

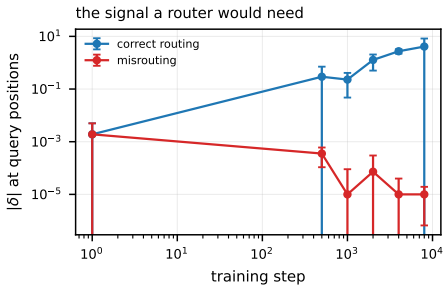

Figure 1: The marginal utility of attention at query positions, $|\delta|$, under two training regimes differing only in the routing mask (5 seeds, log–log). Under correct routing it grows by three orders of magnitude; under misrouting it stays at the floor. The quantity a router would have to estimate in order to escape the deadlock becomes available only after the deadlock is already broken.

A second and sharper point follows from the task's structure. MQAR labels are `IGNORE_INDEX` at every non-query position, which is 94% of them, so at those positions δ_t is not small, it is undefined. The marginal-utility router of Section 3 was therefore not mistuned. Its regression target did not exist almost everywhere it was asked to fit.

It is not a matter of budget headroom. Each sequence carries exactly 16 query positions and the budget is exactly 16, so the oracle mask is the query set and a router is allowed no errors at all. Raising the budget to 1.5, 2 and 4 times the query count leaves learned routing at chance at every level ($0/3$ each, accuracy 0.0002), while q-recall rises in proportion to the budget and tracks what random selection would give, 0.247 against 0.25 expected at $4\times$ (Appendix E). The router has not partially learned and run out of room; extra capacity buys only extra chance coverage.

Nor does the signal appear later. The measurement above covers the window. A natural objection is that a router could simply wait, so we probed the whole of training with three label-free quantities, scoring each by how well it ranks query positions above the rest (Appendix G). Perturbing the output distribution, $\mathrm{KL}(\text{dense} \,\|\, \text{none})$, is uninformative throughout: AUC averages 0.49 across the long pre-acquisition phase under both dense and oracle routing, and once the task is learned it falls *below* chance, 0.22 and 0.13 respectively, ranking query positions beneath the rest. The attention pattern itself *does* carry the signal, entropy 0.46 at query positions against 1.61 elsewhere, AUC 0.93, but not until step $31\,500$ of $32\,000$, over the same interval in which accuracy rises from 0.44 to 0.97. Every probe we have recovers the model's existing reliance on attention, which is a function of the routing it was trained under; the signal becomes readable exactly when the task is already solved.

Consequence: external routing information appears to be necessary. If the internal signal is absent during the window in which routing is decided, an intervention that supplies no external information has nothing to work from. We state this as an inference from the measurement rather than as a theorem, it holds for interventions whose only input is the task loss, and we cannot rule out a signal we did not think to measure. We tested it rather than assuming it, with five interventions spanning the plausible design space, all applied at the window the lock-in result identifies, all at $d_{\text{model}}{=}128$ with the router reading SSM features, 6 seeds each:

The oracle arm is the control that makes the nulls interpretable: the same forward pass, harness and seeds reach 0.986 ± 0.029 when supplied with correct routing, so the sixty failures are not an artifact of the implementation. Each intervention was swept over $K \in \{100, 500, 2000\}$ at 6 seeds.

The scaffold result is the informative one: supplying correct routing *temporarily* does not help, because at handoff the router is still untrained and routing reverts to random. Protecting the backbone and making the router learnable are separate problems, and solving one does not solve the other.

The requirement is threefold, and each part is measured. Section 4 shows the deadlock is asymmetric, with δ at the floor under misrouting but reaching 2.72 under correct routing. That

suggests a bootstrap: route by a heuristic briefly, regress the router onto the now-informative δ, then discard the heuristic. We implemented exactly that, and it fails $0/3$ even with an *uncorrupted* heuristic, in a setting where the same heuristic used permanently succeeds $2/3$. With the corruption sweep, the requirement separates into three conditions, none droppable:

requirement	evidence	runs
external	five interventions supplying no outside signal	0/60
permanent	scaffold and δ-bootstrap, perfect signal, temporary	0/18
	the same bootstrap with a $5\times$ longer scaffold	1/3
near-perfect	the same heuristic at 25% and 50% corruption	0/18
all three	heuristic routing, uncorrupted, permanent	2/3

Permanence is graded, not binary: lengthening the scaffold from 2000 to 8000 steps, past the point where δ separates from noise, moves the bootstrap from $0/3$ with every run at exactly zero to $1/3$ with two partial recoveries. Handing off before δ is informative is what kills the short scaffold, but a scaffold occupying 20% of training is converging on the permanent method it was meant to replace, so the claim worth making, that a brief discardable scaffold suffices, is refuted.

The last row is the sharpest limitation: the intervention that works supplies a near-perfect routing signal and never stops. Real correlates of recall demand are unlikely to meet that bar, so this characterises what is required rather than offering a recipe.

5 Two obstructions, both necessary to remove

Instrument. Stage 1 supervises the router directly against the query positions with the backbone frozen, reaching AUC 0.987 in under a minute; Stage 2 freezes the router and trains the backbone. The router reads only the causal repeat indicator, whether the token has appeared earlier in the sequence, a pure function of the input, and therefore immune to drift in the learned features. Labels enter only at training time. This is a sharper version of Rosenbaum's own suggestion and is not presented as a contribution: its role is to let us supply correct routing at a time of our choosing, which is what Section 6 needs.

Both halves are necessary and neither suffices: drift-immune features without Stage 1 supervision give 0.0002 ± 0.0001, Stage 1 supervision on drift-prone features gives 0.0003 ± 0.0001, and the two together give 0.9485 ± 0.0199 against an oracle of 0.9969 ± 0.0011, every seed of the method above every seed of either ablation, by roughly three thousandfold (Table 2). Its q-recall is 0.857 with no variance across seeds, which looks like an error and is not; Appendix K shows the missing 0.143 is budget competition, not a failure to detect queries.

Feature drift, measured directly. Perturbing the SSM and re-measuring the same frozen router drops q-recall from 0.835 to 0.446 when it reads SSM features alongside the indicator, and leaves it at 0.851 when it reads the indicator alone.

5.1 The anomaly

One seed in the drift-immune-only arm reached q-recall 0.860 and held it for $25\,000$ steps. Its routing was correct and stable for well over half of training, and it still scored 0.0000, the loss sitting at the 8.32 floor throughout. Whatever Stage 1 contributes is therefore not simply correct routing, since straight-through produced correct routing here and it changed nothing. What appears to matter is the ordering, which is what the rest of this paper is about.

6 Irreversible lock-in at initialisation

Protocol. We apply a deficit window of fixed length W at onset t_0, with oracle routing everywhere outside it and total steps held at T. Inside the window, routing is random at full budget, so the attention budget is spent but misallocated. Every run therefore sees the same $T - W$ correctly-routed steps, and all arms are evaluated under oracle routing. We bypass the router entirely and

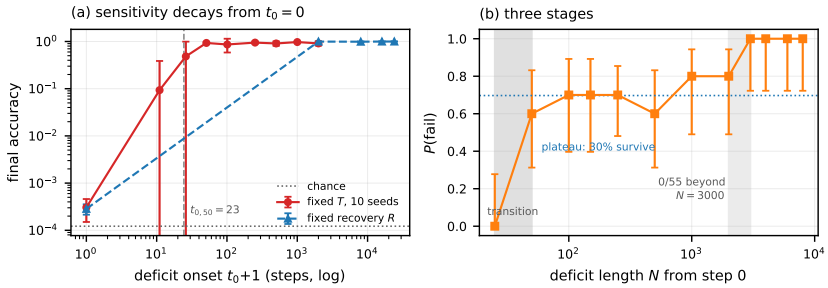

Figure 2: **Routing lock-in at initialisation.** (a) A deficit window of fixed length $W = 8000$ applied at varying onset t_0, with $T = 40\,000$ held constant so every run receives an identical $32\,000$ correctly-routed steps. Note the log-scaled ordinate: delaying the identical window by fifty steps moves the outcome from chance to near-ceiling, and sensitivity decays monotonically from $t_0 = 0$ rather than peaking at an intermediate onset. (b) Deficit applied from step 0 for varying length N, showing three stages: harmless below $\sim$48 steps, a metastable plateau at $\sim$30% survival to $N = 2000$, and universal failure beyond $N = 3000$. Ten seeds per point unless annotated. Chance is 1.22×10^{-4}.

supply masks externally, which isolates the backbone's response to its own attention history from any question about whether a router could be learned. The design is the sliding-window protocol of Achille et al. (2019), who introduce "the deficit in a short window of constant length … starting at different epochs."

Conservative by construction. Holding T fixed makes onset and recovery time anticorrelated, and the confound runs in one direction only: later onsets leave less time to recover, so it predicts that late is worse. Every signature of timing-dependence runs the other way, so any such finding understates the real effect, and the one we obtain, maximal damage at $t_0 = 0$ with full recovery by $t_0 = 50$, is the one it argues hardest against. Only a flat result would be ambiguous, and that is what the equal-recovery control exists to resolve.

The shape is not the one critical periods have. In Achille et al. (2019) the onset curve is non-monotonic: sensitivity is low at $t_0 = 0$, peaks in the central part of the early rapid learning phase, and decays for late onsets. We looked for that shape at the resolution where it would have to appear, nine onsets from 0 to 2000, ten seeds each, and it is absent. Sensitivity is maximal at $t_0 = 0$ and decays monotonically from there.

We therefore do not claim a critical period in the developmental sense, and have named this section accordingly. What the protocol establishes is weaker and more specific: an irreversible commitment made at initialisation, which a brief delay is enough to avoid. That is worth separating from the vision phenomenon it was designed to detect, because the mechanisms need not be the same, and because a reader who takes the analogy too literally will expect a peak that our data do not contain.

The curve is a step, and the step is about twenty-three optimiser updates wide. An eight-thousand-step window of completely misallocated attention is fatal at step zero and harmless if it begins fifty steps later (Figure 2a). Ten seeds at each of nine onsets from 0 to 2000 give survival counts of $0/10$, $1/10$, $5/10$, then $10/10$ at $t_0 = 50$ and at ceiling thereafter (Table 4 lists every run). Fitting a binomial logistic in $\log t_0$ puts the half-way onset at $t_{0,50} = 23.4$ steps, 95% CI $[16.0, 32.7]$ by parametric bootstrap. At $t_0 = 0$ accuracy is 0.000305 ± 0.000155 against a chance rate of 1.22×10^{-4}; by $t_0 = 50$ it is 0.9318 ± 0.0485, and every later onset is indistinguishable from the no-deficit reference (0.9969 ± 0.0007, ten seeds) given seed variance. So the outcome is decided inside the first 0.06% of training, not the 5% an onset grid of $\{0, 2000, 4000, \dots\}$ would suggest.

The recovery confound runs against this twice over: t_0=0 has the longest recovery window in the sweep, $32\,000$ oracle steps, and is the arm that fails, while holding recovery fixed instead of total steps gives the same answer. Nor is this mere degradation, with 8192 evaluation query positions chance predicts one correct answer, and the seeds scored two, zero and two.

The failure is not insufficient training. Learning is heavily back-loaded, under oracle routing accuracy is 0.032–0.066 at step 20 000 and 0.964–0.995 by 32 000, so one might ask whether the $t_0 = 0$ arm simply received too few correctly-routed steps. It did not: a model routed correctly from step 1 reaches 0.964–0.995 after exactly the 32 000 steps that arm receives over steps 8001 to 40 000, and the $t_0 = 2000$ arm receives the same 32 000 and reaches 0.957. The quantity of correct routing is fixed across all three; only its position varies, spanning chance to ceiling.

The window closes before the task is learned at all. Under oracle routing the model scores 0.0000–0.0002 at step 1000 and 0.0066–0.0071 at step 2000 (2 seeds) against a final 0.9969. The interval in which routing becomes irreversible therefore falls where the model has acquired under one per cent of its eventual accuracy. The practical consequence is that this failure cannot be caught by watching a learning curve: by the time performance is measurable, the window has closed.

How little misrouting is enough. Applying the deficit from step 0 for N steps locates the recovery threshold. Outcomes are close to binary, ceiling or chance, so we report survival counts rather than means, which would be meaningless, and resolve the informative region with ten seeds per point.

The transition is localised to fifty steps. Because the outcome is close to binary, the quantity to report is a failure *probability*. Ten seeds per point, across two orders of magnitude in deficit length (Figure 2b):

N	% of training	survived	P(fail) [95% Wilson]
25	0.062%	**10/10**	**0.00** [0.00, 0.28]
50	0.125%	4/10	**0.60** [0.31, 0.83]
100	0.25%	3/10	0.70 [0.40, 0.89]
150	0.375%	3/10	0.70 [0.40, 0.89]
250	0.625%	6/20	0.70 [0.48, 0.85]
500	1.25%	4/10	0.60 [0.31, 0.83]
1000	2.5%	2/10	0.80 [0.49, 0.94]
2000	5%	2/10	0.80 [0.49, 0.94]

Twenty-five misrouted steps are harmless; fifty cost most runs the task. Rather than rest this on two adjacent Wilson intervals failing to overlap — which establishes that the points differ but not where the transition sits — we fit a binomial logistic in $\log N$ with a free lower asymptote, and locate the threshold at $N_{50} = 47.5$ steps, 95% CI $[30.6, 57.0]$.

The lower asymptote is not a modelling convenience, and it carries the one result in this section we did not anticipate. Between $N = 50$ and $N = 2000$ the failure rate does not continue to climb: it plateaus, with 27 of 89 runs surviving (30.3%, 95% CI $[21.8, 40.5]$). A fortyfold increase in deficit length over that range leaves the survival rate flat. Roughly three seeds in ten are simply resistant, and the population is bimodal rather than uniformly damaged.

That plateau is metastable, not a floor. Extending the sweep past it, no run survives at all: 0 of 55 across $N \in \{3000, 4000, 6000, 8000, 16\,000, 32\,000, 40\,000\}$, against 27 of 89 on the plateau (Fisher exact, $p = 3.0 \times 10^{-14}$). Resistance therefore has a limit, and it falls between $N = 2000$ and $N = 3000$.

The commitment is thus made in three stages rather than one: nothing below ~ 48 steps, a partial and seed-dependent commitment that saturates immediately and holds to 2000, and universal failure beyond 3000. A caveat we state rather than hide: two of the plateau runs landed at intermediate accuracy (0.489 at N=250, 0.318 at N=1000), so the basin boundary is not infinitely sharp even within a stage.

A learned router reaches the same state on its own. The protocol supplies masks externally, which isolates the backbone but leaves open the question that matters for Section 3: does a *learned* router fail this way, or is the deficit an artificial condition with no bearing on it? Tracking the attention pathway's RMS relative to the recurrent path over the first 400 steps (3 seeds, $\eta = 3 \times 10^{-3}$), an imposed deficit drives the ratio from 2.12 to 0.0121 and a learned router drives it from 1.93 to 0.0090, the two trajectories are indistinguishable, while oracle routing raises it to 41.9. The state

the deficit creates artificially is the one a learned router arrives at unaided, which is what licenses reading this section as an account of Section 3 rather than as a separate phenomenon.

It is not an artifact of the single-block readout. In the model above everything after the attention readout is position-wise, so logits_t depends on mask_t alone and attention can only help the token it sits on. We therefore repeated the protocol on a stacked model in which attention in every block feeds the next. No single learning rate can test this, the deficit only bites at $\eta \geq 3 \times 10^{-3}$, where four blocks fail to train at all, so we hold the rate the deep model tolerates and raise it only inside the window, with a control that applies the identical rate schedule under *correct* routing (Appendix D). At depth 4 the model trains to ceiling (0.9996), survives the rate schedule alone (0.9998), and fails in four runs of five once the same window is misrouted (0.2000). Varying the window, depth 4 is already saturated at 50 misrouted steps while depth 1 is untouched at 800. The comparison with the $\sim$48-step threshold earlier in this section is not a contradiction but a consequence of the rate: that threshold is measured at $\eta = 3 \times 10^{-3}$ throughout, whereas here the run sits at 1×10^{-3} outside the window, and at the lower rate a single block tolerates far longer deficits (Appendix D). Read at matched settings, the deeper model — the one in which attention can influence other positions — is at least sixteenfold *more* fragile, not less.

The plasticity signature is present in the window that matters. Achille et al. (2019) attribute critical periods to a decay of information plasticity, measured by the Fisher information of the weights. We estimate the Fisher trace of the backbone from single-example gradients under healthy routing (Figure 4).

Two things make the measurement usable, both detailed in Appendix H. The estimate is reproducible only inside the window that matters, two seeds agree to within 1.0–$1.3\times$ up to step $\sim$1200 and diverge past it, so we report the trace only that far and shade the rest, and the decay is not an artifact of the loss falling, since the trace drops by $5.3\times$ and $15.9\times$ across two seeds over steps 0–2000 while the loss moves under 10%, onto a floor it then sits on for tens of thousands of steps. Plasticity collapses by an order of magnitude while the model learns essentially nothing.

The two curves therefore align: the interval over which misrouting is unrecoverable is the interval over which Fisher information collapses. We report this as a coincidence of two independently measured quantities, not as evidence that one causes the other.

7 A MECHANISM, AND A PREDICTION IT FAILS

In a linear surrogate of the two-pathway block, misrouted attention is uncorrelated with the target, so the readout on the attention path receives nothing but decay and the attention parameters inherit the same factor. Suppression is therefore active rather than passive, and self-locking: once the readout is small the parameters stop receiving gradient at all. That accounts for the irreversibility and for the plateau between N=50 and N=2000, though not for the second collapse past N=3000. We measure it directly, at $\eta = 3 \times 10^{-3}$ the attention pathway's magnitude relative to the recurrent path falls from 2.12 to 0.0121 under misrouting while rising to 41.9 under correct routing, the two regimes ending a factor of $\sim 3.5 \times 10^3$ apart at the lower rates and $\sim 1.4 \times 10^6$ apart at 6×10^{-3} (3 seeds each).

The account also predicts $N_{\text{crit}} \propto 1/\eta$. We pre-registered that and it fails, in three separate places: $N_{\text{crit}}\,\eta$ varies more than eightfold against a threshold of two, the obvious repair (that $\mathbb{E}[\alpha^2]$ scales with η) is refuted by measurement, and the suppression timescale is independent of η altogether. We therefore report the mechanism as qualitative and leave the scaling open; Appendix B has the derivation, the sweeps and the failures.

8 RELATED WORK

The problem and the fix are older than this paper. Rosenbaum et al. (2019) name the "chicken-and-egg" problem directly, at initialisation the router must learn the value of modules that "have not yet taken on any real meaning", and propose the remedy of freezing them briefly or lowering their learning rate. Our two-stage instrument is an instance of that family. What their account does not

predict is Section 5.1: a router correct and stable for $25\,000$ steps that still fails. The correction is that the problem is not failing to stabilise but that stabilising late is worthless, which is a claim about timing that no work in this area makes.

Gradient flow, and lock-in observed but not manipulated. The dominant explanation is that top-k selection is discontinuous, so unselected candidates receive no gradient toward selection, which motivates differentiable relaxations and shapes Mixture-of-Depths (Raposo et al., 2024); sparse mixtures also degenerate when a few experts capture all traffic (Chi et al., 2022). Both concern whether gradient reaches the router, and Section 3 shows that is not sufficient. Separately, Xue et al. (2024) and Muennighoff et al. (2024) report routing fixed within the first 1% of pretraining, our regime, but neither manipulates timing, so neither can say what that costs. Achille et al. (2019) establish that an early deficit does irreversible damage where the same deficit later is harmless; we adopt their sliding-window protocol unchanged, and to our knowledge it has not been applied to conditional computation. Our setting, token-level routing between recurrent and attention paths, follows Lieber et al. (2024) and Ren et al. (2025); Appendix C has the quotations in full.

9 Scope and limitations

Robustness, capacity, and protocol. The method needs a near-perfect signal: corrupting the repeat indicator at 10% collapses it to chance under two of three corruption types, and the third is bimodal rather than partially robust (Appendix L). The regime is bounded above as well, at $d_{\text{model}} = 512$ the SSM alone reaches 0.712 ± 0.315, so the task has largely stopped being attention-limited, and where attention is not load-bearing there is little for any routing policy to decide. One limit is specific to the protocol: we supply the deficit masks externally, and although the suppression trajectories show a learned router reaching the same state, we never measure the window with router and backbone training jointly.

Real text: a bound, and a control on benchmark choice. We searched WikiText-103 for a configuration in which attention is load-bearing, scoring on the *associative-recall slice*: positions whose current token occurred earlier and whose continuation there is the token now due. Across four one-block configurations the headroom was $+0.83\%$, -0.82%, $+1.41\%$, -0.18%, underpowered rather than null, with seed noise at 1.69% relative and a minimum detectable effect of $\sim 3\%$ at $n = 3$. An effect above that would have been detected with high probability; smaller ones are not excluded. That suffices, since routing only redistributes the gap between no-attention and dense attention, and a gap under 3% leaves no policy anything measurable to compete over; a non-causal oracle spending the budget exactly where recall would pay gains $+0.7\%$, inside noise. We keep this rather than drop it: it bounds where token-level allocation can pay, and shows we looked on natural data before studying a task built to contain the effect. Appendix J gives the two confounds eliminated.

10 What this leaves for method design

The constraint we end with is narrower than an impossibility result and more useful than a negative one. Routing cannot be learned from the task in the interval where it must be decided: the quantity that would supply the gradient is zero there, and every substitute we constructed reports the routing the model was given rather than the routing it needs. A method that works must import routing information from outside the task loss and keep importing it, brief scaffolds fail not for being brief, but because the moment they stop, the router is again learning from a signal that does not exist.

Two openings remain that we could not close: making attention competent at initialisation, so δ is non-degenerate inside the window rather than after it, and finding a routing correlate that is cheap, external, and robust to corruption, since the heuristic that works here dies at 10% noise. Both seem worth more than another optimiser.

Ethics Statement

This work trains small models on a synthetic associative-recall task and on WikiText-103, a public corpus of encyclopedia text. No human subjects, personal data, or user-generated content were involved, and no new data was collected. The models are far too small to be deployed or misused, and the findings are negative results about the trainability of routers rather than a capability advance. The clearest practical implication is a caution: systems that route conditionally may be committing to allocations within the first fraction of a percent of training, which is a failure mode worth knowing about rather than a risk this work creates.

Reproducibility Statement

Every number in this paper is produced by a script in the code submitted as supplementary material, writing to a CSV read directly by the figure generator; no figure is drawn from a hand-transcribed number, and a checking script re-derives each headline value from its CSV and diffs it against the manuscript. The task generator is a faithful port of the Zoology MQAR generator (Arora et al., 2023), with the same algorithm and defaults; the only change is that it draws from a local random number generator rather than NumPy's global one. The recurrent path is upstream `mamba-ssm` with its compiled CUDA kernel, not a reimplementation. All multi-seed results report individual runs alongside the mean so a single outlier cannot hide inside an average; bimodal results are reported as runs only. All experiments run on a single consumer GPU.

Every arm is evaluated on the same held-out batch of 512 sequences, which carries 8192 query positions. Two consequences are worth stating plainly. The sampling error of a single accuracy estimate is binomial and small, ± 0.0024 at an accuracy of 0.95 and ± 0.00019 near chance, which is an order of magnitude below the seed-to-seed spread we report (e.g. 0.0485 at $t_0{=}50$), so seed variance, not evaluation noise, dominates every interval in this paper. And because the batch is shared across arms, it cannot manufacture a difference between them; a draw that happened to be unrepresentative would shift all arms together rather than separate them. We checked this rather than resting on the arithmetic: re-evaluating trained models on twelve independent batches gives 0.9971 ± 0.0007 under oracle routing and 0.0003 ± 0.0002 after an early deficit, against fixed-batch values of 0.9968 and 0.0005.

Runs are seeded but not bit-reproducible: the fused attention backend uses a non-deterministic algorithm, so two executions of an identical schedule differ slightly. We observed this directly, the onset arm $t_0{=}0$ and the duration arm $N{=}8000$ are the same schedule under the same seeds and returned $(0.000244, 0.000000, 0.000244)$ and $(0.000244, 0.000366, 0.000244)$. At chance this is immaterial, and every claim here is stated at a scale far above it.

Use of Large Language Models

Used to assist with English language editing, grammatical corrections, and proofreading to improve clarity. The model was not used to generate original research concepts, draft sections of the paper, or perform data analysis.

References

Alessandro Achille, Matteo Rovere, and Stefano Soatto. Critical learning periods in deep networks. In *International Conference on Learning Representations (ICLR)*, 2019. arXiv:1711.08856.

Simran Arora, Sabri Eyuboglu, Aman Timalsina, Isys Johnson, Michael Poli, James Zou, Atri Rudra, and Christopher Ré. Zoology: Measuring and improving recall in efficient language models. *arXiv preprint arXiv:2312.04927*, 2023.

Zewen Chi, Li Dong, Shaohan Huang, Damai Dai, Shuming Ma, Barun Patra, Saksham Singhal, Payal Bajaj, Xia Song, Xian-Ling Mao, Heyan Huang, and Furu Wei. On the representation collapse of sparse mixture of experts. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. arXiv:2204.09179.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Opher Lieber, Barak Lenz, Hofit Bata, Gal Cohen, Jhonathan Osin, Itay Dalmedigos, Erez Safahi, Shaked Meirom, Yonatan Belinkov, Shai Shalev-Shwartz, et al. Jamba: A hybrid transformer-mamba language model. *arXiv preprint arXiv:2403.19887*, 2024.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Pete Walsh, Oyvind Tafjord, Nathan Lambert, et al. OLMoE: Open mixture-of-experts language models. *arXiv preprint arXiv:2409.02060*, 2024.

Destiny Okpekpe and Antonio Orvieto. Revisiting associative recall in modern recurrent models. *arXiv preprint arXiv:2508.19029*, 2025.

David Raposo, Sam Ritter, Blake Richards, Timothy Lillicrap, Peter Conway Humphreys, and Adam Santoro. Mixture-of-depths: Dynamically allocating compute in transformer-based language models. *arXiv preprint arXiv:2404.02258*, 2024.

Liliang Ren, Yang Liu, Yadong Lu, Yelong Shen, Chen Liang, and Weizhu Chen. Samba: Simple hybrid state space models for efficient unlimited context language modeling. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2406.07522.

Clemens Rosenbaum, Ignacio Cases, Matthew Riemer, and Tim Klinger. Routing networks and the challenges of modular and compositional computation. *arXiv preprint arXiv:1904.12774*, 2019.

Fuzhao Xue, Zian Zheng, Yao Fu, Jinjie Ni, Zangwei Zheng, Wangchunshu Zhou, and Yang You. OpenMoE: An early effort on open mixture-of-experts language models. In *International Conference on Machine Learning (ICML)*, 2024. arXiv:2402.01739.

A Experimental detail

Task. Zoology MQAR: vocabulary 8192, sequence length 256, 16 key–value pairs, query positions controlled by a window parameter. Evaluation uses a fixed batch of 512 sequences (seed $987\,654\,321$), giving 8192 query positions and an expected 1.0 correct predictions at chance.

Model. $d_{\text{model}} = 128$ unless stated; Mamba with $d_{\text{state}} = 16$, $d_{\text{conv}} = 4$, expansion 2; causal attention with $d_{\text{model}}/32$ heads; FFN with expansion 2 and GELU. Attention output is added to the SSM output at routed positions.

Optimisation. AdamW, weight decay 0.01, gradient-norm clipping at 1.0, batch size 32. Stage-2 learning rate 3×10^{-3}; Stage-1 router supervision 10^{-3} for 3000 steps. Headline runs are $40\,000$ steps.

Critical-period protocol. The router is bypassed and masks supplied externally, so the measurement isolates the backbone's response to its attention history. The deficit mask is drawn uniformly at random subject to spending exactly the same budget as the oracle mask (16 of 256 positions), so the two differ in *placement* only. Evaluation always uses the oracle mask, making the mask the model saw during training the only difference between arms.

B Why the commitment is irreversible

The experiments establish *that* early misrouting is unrecoverable. This section gives a mechanism, in a linear surrogate of the two-pathway architecture, and derives from it a quantitative prediction that Section I tests.

Setup. Write the block's output as a sum of a recurrent and a routed-attention contribution,

$$\hat{y} \;=\; w_s^\top s \;+\; w_a^\top \alpha, \qquad \alpha \;=\; m \odot a, \tag{1}$$

where s is the recurrent feature, a the attention feature produced by parameters θ_a, m the routing mask, and w_s, w_a readout weights. Take a squared loss $L = \frac{1}{2}\mathbb{E}[(\hat{y}-y)^2]$. During a deficit the mask is drawn independently of the data, so the routed attention output carries no information about the target: $\mathbb{E}[y\,\alpha] = 0$.

Proposition 1 (self-reinforcing suppression). *Under the deficit, the attention readout decays exponentially at a rate set by the attention output's own second moment, and the gradient reaching the attention parameters is suppressed by the same factor. Explicitly, with learning rate η and $\kappa \triangleq \eta\,\mathbb{E}[\alpha^2]$,*

$$w_a(N) \;=\; w_a(0)\,e^{-\kappa N}, \qquad \big\|\nabla_{\theta_a} L\big\| \;\le\; |w_a(N)| \cdot C \;=\; O\big(e^{-\kappa N}\big)\,. \tag{2}$$

Proof sketch. The readout gradient is $\partial L/\partial w_a = w_a\,\mathbb{E}[\alpha^2] - \mathbb{E}[y\alpha]$, and the second term vanishes by construction, leaving pure decay $\dot{w}_a = -\eta\,\mathbb{E}[\alpha^2]\,w_a$. The parameter gradient factors as $\nabla_{\theta_a} L = w_a\,\mathbb{E}[(\hat{y}-y)\,\nabla_{\theta_a}\alpha]$, so it inherits the factor w_a. □

Two consequences matter. First, the decay is *active*, not passive: it is driven by $\mathbb{E}[\alpha^2]$, the variance of the attention pathway's own output, not by weight decay. The deficit does not merely fail to teach the model to use attention—it teaches the model to *ignore* attention, because ignoring a noisy input is locally optimal. Second, the lesson is self-locking. Once $w_a \approx 0$ the attention parameters stop receiving gradient, so θ_a freezes at whatever value it held; and recovery requires re-growing w_a, whose growth rate is proportional to $\mathbb{E}[y\alpha]$, which requires θ_a to still encode something useful. Neither factor can move without the other. The point $w_a = 0$ is stationary in the joint dynamics, and the deficit drives the system into its basin.

A quantitative corollary, which we test and which fails. Equation (2) suggests the number of deficit steps needed to suppress the pathway by a factor e is

$$N_{\text{crit}} \;=\; \frac{1}{\kappa} \;=\; \frac{1}{\eta\,\mathbb{E}[\alpha^2]}, \tag{3}$$

so if $\mathbb{E}[\alpha^2]$ were constant the critical window would scale as $1/\eta$. Section I tests this and reports that it does not hold, in either its original form or under the obvious repair. We state the corollary anyway, because a mechanism that makes no quantitative commitment cannot be wrong, and one that can be is worth more even when it is.

C Related work, in full

Routing trainability. Rosenbaum et al. (2019) state the problem exactly: at initialisation "the router needs to learn the value of the modules, while these are effectively randomly trained and have not yet taken on any real meaning," and the resulting "chicken-and-egg" problem "can result in the routing dynamics never stabilizing, leaving the network at effectively random performance." Their proposed remedy is "fixing their parameters for a short initial period, or by reducing their learning rate." We claim neither the problem nor that family of fix.

Why Mixture-of-Depths does not meet this. Raposo et al. (2024) note that "the top-k operation is non-causal," depending on routing weights of later tokens, and train the deployable router on labels derived from that non-causal top-k. Their training-time router is therefore approximately correct by construction and never enters the deadlock. Their own explanation, "the ease of learning this prediction problem," with upwards of 97% accuracy early in training, is a direct statement that *supervised* routing is easy while *task-loss* routing is hard, which is the separation Section 5 isolates.

Early lock-in at scale. Xue et al. (2024) report "Early Routing Learning": token-to-expert assignments fixed early and largely unchanged after. Muennighoff et al. (2024) quantify it, after 1% of pretraining $\sim$60% of top-8 routing already matches the final checkpoint, rising to $\sim$80% by 40%,

against 12.5% for random routing. Both *observe*; neither manipulates timing. OpenMoE does attribute harm, but through late-sequence tokens dropped by saturated experts, which is a claim about inference rather than about irreversibility. That gap is what Section 6 occupies. We test no MoE model.

D DEPTH, AND WHY ONE LEARNING RATE CANNOT TEST IT

Every result in the body is measured on a single block, where everything after the attention readout is position-wise and logits_t therefore depends on mask_t alone: attention can only help the token it sits on. Stacking blocks removes that restriction, because block 1's attention rewrites the residual stream that block 2's recurrent path mixes across positions. Whether lock-in is a property of routing or of the single-block readout is the question a reader is most entitled to ask, so we tested it.

The obstacle. Testing depth needs the deficit to bite *and* the deep model to train, and those pull in opposite directions in the learning rate:

learning rate	depth 1, t_0=0 (want failure)	depth 4, no deficit (want ceiling)
3×10^{-3}	**0/5**	0/5 (0.021)
2×10^{-3}	1/3	2/3
1.5×10^{-3}	1/3	**3/3** (0.9991)
1×10^{-3}	5/5 (no damage)	**3/3** (0.9999)

The requirements never overlap. At 2×10^{-3}, the one candidate rate, both are bimodal rather than compromising. A first attempt that ignored this reported depth-2 and depth-4 nulls whose *no-deficit controls had themselves failed* ($1/5$ and $0/5$); those numbers measure an optimisation failure, not a routing result, and we discard them.

Decoupling plasticity from trainability. Since the mechanism attributes damage to gradient magnitude while misrouting is present, we train at the rate the deep model tolerates (1×10^{-3}) and raise it to 3×10^{-3} only inside the deficit window. Two arms would not separate the causes, so there are three: `no-deficit` (flat rate, correct routing), `switch-only` (the identical rate schedule, correct routing), and `deficit` (the schedule plus misrouting). The window is W=200 rather than 8000 because the threshold measured in Section 6 is $\sim$48 steps, so 200 carries fourfold margin while exposing the deep model to $40\times$ less high-rate training, which is what keeps the `switch-only` control interpretable.

	no-deficit	switch-only	deficit
depth 1	5/5 0.9983	5/5 0.9986	5/5 0.9983
depth 4	5/5 0.9996	5/5 0.9998	**1/5** 0.2000

At depth 4 the model trains to ceiling, survives the identical learning-rate schedule under correct routing, and fails in four runs of five as soon as the same window is misrouted. Lock-in is therefore not an artifact of the single-block readout.

Dose-response, and a reversal. Varying the window at both depths:

deficit length	$W = 50$	$W = 200$	$W = 800$
depth 1	5/5	5/5	5/5
depth 4	1/5	1/5	2/5

`switch-only` is $5/5$ in every cell at both depths. Depth 4 is already saturated at W=50 while depth 1 is untouched at W=800, so the thresholds differ by at least sixteenfold in the direction opposite to the one we expected: the deeper model, in which attention can influence other positions, is the *more* fragile of the two. We report the ratio as a bound because we did not sweep below W=50 at depth 4, so its threshold may be lower still.

Context length: attempted, not answered. We tried the same protocol at 1024 and 2048 tokens, twice. Scaling the number of key-value pairs with the context holds query density fixed but also scales the memory demand, and the no-deficit control fails ($0/3$). Holding the pairs fixed at 16 so that only the haystack grows fails as well ($0/3$ at both lengths), so the obstacle is the context length itself at this model size rather than the recall load. Since the control does not train, neither sweep can say anything about the deficit, and we report no result for context length rather than a null.

What we could not test. A two-block variant with attention in the first block only, the cheapest instantiation of cross-position influence, does not train at these settings (no-deficit $2/5$, 0.3937), so every number from that configuration is uninterpretable and we draw nothing from it.

Table 2: Ablation, 3 seeds. Every seed of the method exceeds every seed of every ablation.

Configuration	Accuracy	Individual runs
Drift-immune features, no Stage 1	0.0002 ± 0.0001	0.0002, 0.0000, 0.0002
Stage 1 supervision, drift-prone features	0.0003 ± 0.0001	0.0002, 0.0004, 0.0002
Both (the method)	$\mathbf{0.9485 \pm 0.0199}$	0.9308, 0.9447, 0.9700
Oracle	0.9969 ± 0.0011	0.9961, 0.9965, 0.9982

E DOES THE BUDGET HAVE ENOUGH SLACK?

The budget is 16 positions and each sequence carries exactly 16 query positions, so the oracle mask *is* the query set and a router is allowed no errors at all: one wrong selection displaces one query. Production mixture-of-experts systems instead run a capacity factor above one, and every token reaches some expert. A reader may reasonably suspect that learned routing fails here because the budget has no headroom rather than because of anything about timing.

We varied the budget from one to four times the query count, at $40\,000$ steps and 3 seeds:

budget	16 (1×)	24 (1.5×)	32 (2×)	64 (4×)
straight-through	0/3	0/3	0/3	0/3
accuracy	0.0002	0.0002	0.0002	0.0002
q-recall	0.064	0.094	0.122	0.247
marginal-utility	0/3	0/3	0/3	1/3

Learned routing is at chance at every budget. The q-recall row is the informative one: it rises in proportion to the budget and matches what random selection would achieve ($64/256 = 0.25$ at $4\times$), so the router has not partially learned and run out of room, it has learned nothing, and additional capacity buys only additional chance coverage. The marginal-utility regressor reaches $1/3$ at $4\times$ on the strength of a single seed at 0.593, which we report rather than round away. Budget headroom is not the obstruction.

F MARGINAL UTILITY OF ATTENTION, FULL NUMBERS

Section 4 quotes the step-4000 row; the full sweep is below.

Table 3: Marginal utility of attention at query positions, 5 seeds. Under misrouting it is identically zero with vanishing variance; under correct routing it grows by orders of magnitude. The signal a router needs in order to escape appears only *after* routing is already correct.

step	δ under random routing	δ under oracle routing
500	-0.0004 ± 0.0002	0.293 ± 0.414
1000	-0.0000 ± 0.0001	0.229 ± 0.181
2000	-0.0001 ± 0.0002	1.274 ± 0.772
4000	$\mathbf{-0.0000 \pm 0.0000}$	$\mathbf{2.722 \pm 0.514}$
8000	-0.0000 ± 0.0000	4.104 ± 4.182

G FOUR PROBES FOR AN INTERNAL ROUTING SIGNAL

A router needs some readable quantity that is larger at positions where attention pays. We tried four, and record them here because the way each fails is more informative than the fact that it does. All are scored by AUC as a discriminator of query positions, which is what a top-k router consumes: ordering, not magnitude.

1. Marginal utility, δ_t. Undefined where it is needed. On MQAR the labels are sparse, so δ_t exists only at query positions and has no negative class to rank against, our first attempt returned AUC $=$ nan. Distilling it would be circular: knowing where δ is defined is knowing the answer.

2. Output perturbation in a densely trained model. We replace the loss difference with a label-free one, $\delta_t^{\mathrm{KL}} = \mathrm{KL}(p_t(\text{dense}) \,\|\, p_t(\text{none}))$, defined at every position. Under dense training AUC averages 0.49 over the first $28\,000$ steps, while accuracy is still under 0.06: a model given attention everywhere learns to rely on it everywhere, so the probe destroys the selectivity it is trying to measure. As the task is finally acquired the AUC does not rise but falls, reaching 0.22 at step $32\,000$ and 0.94 accuracy.

3. Output perturbation in a correctly-routed model. The same quantity in an oracle-trained model gives AUC 0.13, with KL five times *larger* at non-query positions. Off-distribution masking is part of the explanation, under oracle routing the model never had attention off the query positions, so switching it on there is a perturbation it never trained under, but it cannot be all of it, because the dense-trained model, which had attention everywhere, also ends below chance. Whichever mechanism dominates, the direction is the opposite of the one a router needs, and inverting the probe would rank by where attention already is rather than where it would help.

4. The attention pattern itself. This one avoids both traps: it toggles nothing and reads the learned circuit directly. Per position we take the entropy of the attention distribution and the largest weight assigned to a non-adjacent token, on the reasoning that retrieval looks like a sharp peak on a distant key. It works, eventually. AUC sits at 0.45–0.51 for 98% of training and then reaches 0.93 at step $32\,000$, with attention entropy at query positions falling to 0.458 against 1.613 elsewhere. The crossing coincides with task acquisition: accuracy over the same interval goes from 0.44 to 0.97.

What the four have in common. Each probe returns the model's current reliance on attention, and that reliance is a function of the routing policy the model was trained under, uniform under dense routing, confined to the mask under oracle routing, absent under misrouting, and selective only once the task is solved. We state this as the strongest form of the obstruction we can support: we have no policy-independent measurement of where attention is needed, and the one quantity that does become selective becomes so too late to route with.

H FISHER TRACE: ESTIMATOR RELIABILITY

The Fisher trace is estimated from single-example gradients under oracle routing. Reliability is not uniform over training, and the paper's use of it depends on where it is trustworthy.

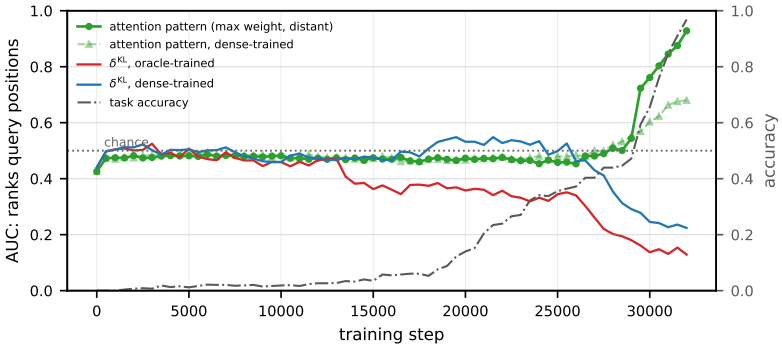

Figure 3: **When an internal routing signal becomes readable.** AUC of each probe as a discriminator of query positions, against training step, with task accuracy overlaid. Every probe sits at chance for the great majority of training. The attention pattern eventually separates the query positions, but only over the same interval in which the task is acquired, so the signal arrives with the solution rather than in time to produce it. δ^{KL} in an oracle-trained model drifts *below* chance, because switching attention on where it was never trained is an off-distribution perturbation, so the probe reports where attention was routed rather than where it helps.

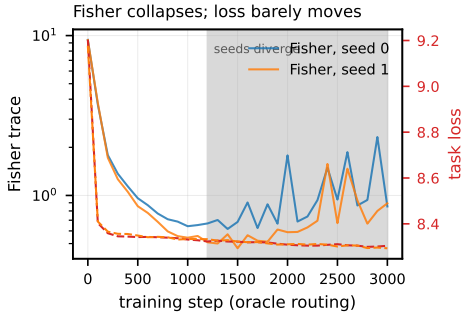

Figure 4: Fisher trace (solid, log scale, one line per seed) and task loss (dashed) under healthy oracle routing. The trace falls by an order of magnitude while the loss moves less than 10%. Shading marks the region beyond step 1200 where the two seeds diverge and the estimator is no longer trustworthy; the critical window lies entirely to its left.

With 128 single-example gradients per probe, two independent seeds agree to within 1.0–$1.3\times$ for all steps up to $\sim$1200. Past that they diverge, reaching $3\times$ by step 2000 and over $1000\times$ once the model converges, which is why Figure 4 shades everything beyond step 1200 and why no claim is made there. A coarser estimate (32 gradients per probe) is dominated by seed variance everywhere and supports nothing at all; the resolution matters more than the range.

The lock-in window established in Section 6 is tens of steps, so it lies far inside the reliable region, which is the only reason the two measurements can be compared.

A decay in the trace is uninformative on its own, because the Fisher trace shrinks mechanically as the loss falls. Over steps 0 to 2000 the trace falls by $5.3\times$ (seed 0) and $15.9\times$ (seed 1) while the task loss falls from 9.20 to 8.31, under 10%, and onto a floor the model occupies for tens of thousands of further steps. The decay therefore outruns the loss by more than an order of magnitude.

I TESTING THE PREDICTION: A PRE-REGISTERED FAILURE

Define N_{crit} as the deficit length at which $P(\mathrm{fail})$ crosses $1/2$. At $\eta = 3 \times 10^{-3}$ Section 6 gives $N_{\mathrm{crit}} \approx 46$, hence $N_{\mathrm{crit}}\,\eta \approx 0.138$; Equation 3 says that product is constant, so we predicted 92 at

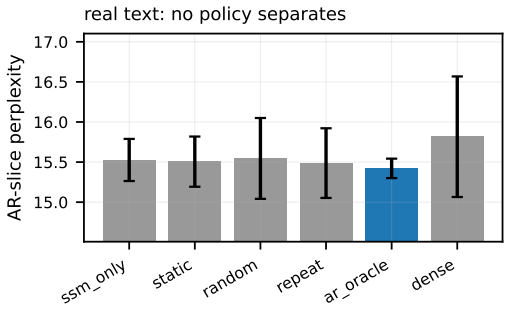

Figure 5: Real text, matched budget, 3 seeds, AR-slice perplexity. No allocation policy separates from any other, including a non-causal oracle that spends the budget exactly where recall would pay. Error bars are one standard deviation across seeds.

1.5×10^{-3} and 23 at 6×10^{-3}, with a factor of two in the product fixed in advance as the falsification threshold.

The prediction fails in both directions, and not only in magnitude. Sweeping $N \in \{12, \ldots, 400\}$ at both rates places neither crossing inside the grid: beyond 400 at the lower rate and below 12 at the higher. The clearest comparison needs no interpolation: a 12-step deficit, 0.03% of training, leaves every run intact at $\eta = 1.5 \times 10^{-3}$ and destroys every run at 6×10^{-3}. So $N_{\text{crit}}\,\eta$ ranges from > 0.60 through 0.138 to < 0.072, a spread beyond $8\times$ against a pre-registered threshold of two.

The more damaging discrepancy is qualitative. Corollary 1 describes a curve that *shifts* along N with its shape preserved, so the asymptotic failure rate should be unchanged. It is not: at $\eta = 3\times10^{-3}$ the failure probability plateaus at $\approx$0.70, while at 1.5×10^{-3} it plateaus at $\approx$0.12 and stays there—$2/16$ at N=200 and $1/10$ at N=400, both with 95% intervals excluding $1/2$. Halving the learning rate does not merely postpone the commitment; it makes catastrophic commitment *rare*. No horizontal rescaling of a fixed curve can produce that, so the failure is of the model's form and not only of its constant.

The obvious repair also fails, and by measurement rather than argument. Equation 3 treats $\mathbb{E}[\alpha^2]$ as constant; if instead $\mathbb{E}[\alpha^2] \propto \eta$ the result would be exactly the observed $1/\eta^2$. Measured directly under the deficit, however, $\mathbb{E}[\alpha^2]$ grows as $\sim\eta^{2.5}$, implying $1/\eta^{3.5}$—overshooting by as much as the constant assumption undershoots, robustly across four averaging windows (Appendix N). We report this rather than fit a third hypothesis.

What survives is this. Proposition 1's qualitative content is untouched, and still explains what the gradient-flow account does not: why the damage is irreversible, and why the failure probability is flat above $N = 50$ rather than growing with deficit length. What fails is turning that mechanism into a scaling law. Empirically the learning-rate dependence is stronger than any account here predicts—N_{crit} moves from beyond 200 steps to under 12 across a fourfold change in η, and explaining it is the clearest open problem this work leaves.

J Real-text controls: learning rate and depth

Learning rate is not the explanation, though it had to be checked. Okpekpe & Orvieto (2025) report that "the choice of learning rate plays a critical role in the performance of modern recurrent models." We measure a 24% swing in AR-slice perplexity across learning rates for the no-attention arm alone (13.65 at 3×10^{-3} against 17.99 at 10^{-2}), an order of magnitude larger than the headroom effects being measured, so a mistuning that affected the arms unequally could have manufactured the entire result. It does not: sweeping each arm independently, both peak at the *same* learning rate (3×10^{-3}), so the shared-rate comparison is unbiased. At their common optimum the AR-slice gap is -1.7%, against $+1.6\%$ at the rate used in the original sweep. The sign is not stable and both magnitudes are within the 1.69% seed noise, which is the substantive finding: at one block the gap is not merely small but not reliably signed.

Depth does not create headroom either. A one-block probe might be unable to express induction, since in-context recall is classically a two-layer construction, so we swept blocks $\in \{1, 2, 4\}$ against training length $\in \{6000, 30\,000\}$ at the common best learning rate (Appendix O). Measured naively, headroom *grows* with depth at 30 000 steps ($+1.4\%, +5.0\%, +5.8\%$), which would overturn the null. It does not: the growth comes from the no-attention arm *degrading* rather than attention helping, and every 30 000-step run is worse than its 6000-step counterpart, because at batch 4 and length 1024 that is $\sim$123M tokens over a corpus of comparable size. At each arm's best configuration the difference is not reliably signed: across two seeds attention loses by 4.2% on one and wins by 1.9% on the other.

No configuration we tested shows attention reliably paying on real text. We report the spurious trend because a reader sweeping only 30 000-step runs at one seed would find a clean, monotone, entirely false result.

K WHY Q-RECALL IS 0.857 ± 0.000

A quantity reported to three decimals with no variance across seeds looks like a mistake, so it is worth saying why it is not. In the repeat-only configuration the frozen router's input depends only on the token sequence, and evaluation uses a fixed batch, so the resulting mask does not depend on the trained weights at all. Being identical across seeds is the correct behaviour, and it is a more direct demonstration of drift-immunity than the perturbation above.

The missing 0.143 comes from budget competition rather than from any failure to detect queries. All 16 true queries carry a repeat flag, so feature recall is 1.000; but a sequence carries 19.49 flags on average against a budget of 16, which means about 3.49 flagged positions get discarded. The same mechanism explains the corruption asymmetry we report in Section 9. Figures here use index-order tie-breaking, giving 0.8571; random tie-breaking gives 0.8292.

L NOISE ROBUSTNESS

Noise robustness. The method requires a near-perfect routing signal. Corrupting the repeat indicator at 10% collapses it to chance under `flip` (0.0004) and `distract` (0.0004), against a clean baseline of 0.941 ± 0.010. `drop` is *bimodal*—individual runs $(0.0005, 0.9823)$, a coin flip rather than partial robustness, and we report the runs rather than their mean. The asymmetry is mechanistic: `drop` only removes true positives while leaving surviving signal clean, whereas `flip` and `distract` inject false positives that occupy the fixed budget and starve genuine queries.

M ONSET SWEEP, FULL NUMBERS

Table 4: Onset sweep, $W = 8000$, $T = 40\,000$, 10 seeds per row. Chance is 1.22×10^{-4}. First five runs shown; a run counts as survived at accuracy > 0.5.

Onset t_0	Deficit covers	Accuracy	Survived	First five runs
0	steps 1–8000	$\mathbf{0.0003 \pm 0.0002}$	**0/10**	0.0002, 0.0004, 0.0002, 0.0006, 0.0002
10	steps 11–8010	0.0932 ± 0.2937	1/10	0.0004, 0.0001, 0.9290, 0.0002, 0.0002
25	steps 26–8025	0.4829 ± 0.5096	5/10	0.0005, 0.0000, 0.8962, 0.0002, 0.9537
50	steps 51–8050	$\mathbf{0.9318 \pm 0.0485}$	**10/10**	0.8779, 0.9841, 0.9070, 0.9568, 0.9058
100	steps 101–8100	0.8644 ± 0.2800	9/10	0.9528, 0.9974, 0.9646, 0.9812, 0.8644
250	steps 251–8250	0.9476 ± 0.0484	10/10	0.9962, 0.9608, 0.9946, 0.9565, 0.9202
500	steps 501–8500	0.9083 ± 0.0985	10/10	0.9236, 0.8333, 0.9069, 0.9971, 0.9574
1000	steps 1001–9000	0.9730 ± 0.0223	10/10	0.9723, 0.9963, 0.9438, 0.9972, 0.9491
2000	steps 2001–10000	0.9105 ± 0.0921	10/10	0.9968, 0.7408, 0.9368, 0.8718, 0.9946
No deficit	—	0.9969 ± 0.0007	10/10	0.9962, 0.9966, 0.9977, 0.9978, 0.9962

Table 4 lists the sweep plotted in Figure 2a. Every arm receives an identical 32 000 correctly-routed steps; only the position of the deficit window varies.

N LEARNING-RATE SCALING: FIGURES AND MEASURED $\mathbb{E}[\alpha^2]$

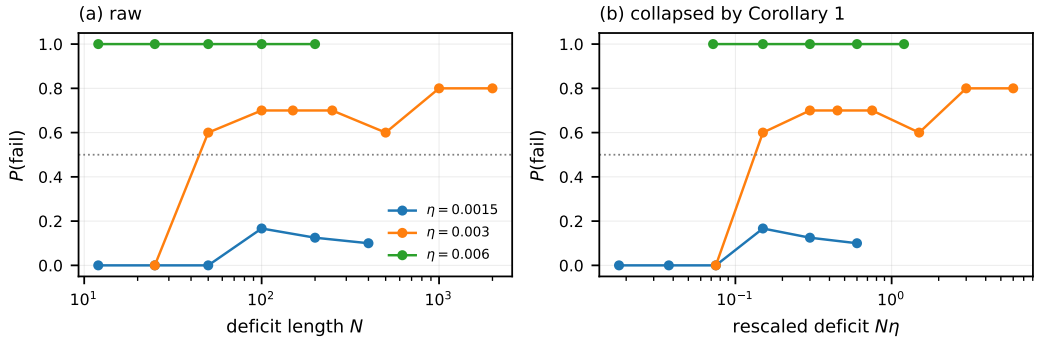

Figure 6: Failure probability against deficit length at three learning rates. **(a)** raw: the curve shifts dramatically, from a crossing beyond $N = 200$ at $\eta = 1.5 \times 10^{-3}$ to below $N = 12$ at 6×10^{-3}. **(b)** the same data with the abscissa rescaled by η. If Corollary 1 held, $N_{\text{crit}}\eta$ would be constant and the three curves would collapse onto one. They do not, which is the falsification of Section I in one picture.

Failure probability at the two additional rates (6 seeds per point).

N	12	25	50	100–200
$P(\text{fail})$ at $\eta = 1.5 \times 10^{-3}$	0/6	0/6	0/6	1/6
$P(\text{fail})$ at $\eta = 6.0 \times 10^{-3}$	**6/6**	6/6	6/6	6/6

window (steps)	12	25	50	200
$\mathbb{E}[\alpha^2]$ ratio over $4\times$ in η	25.1	40.8	36.3	34.8
implied exponent	3.32	3.67	3.59	3.56

Mean squared routed attention contribution under random routing, averaged over the first w steps, 3 seeds per rate, for $\eta \in \{1.5, 3, 6\} \times 10^{-3}$. The ratio is taken across the fourfold range in η. The implied exponent is the x in $N_{\text{crit}} \propto 1/\eta^x$ that Equation 3 would give with the measured $\mathbb{E}[\alpha^2]$; the observed exponent is ≈ 2.

O DEPTH SWEEP

Table 5: Depth sweep, AR-slice perplexity, both arms at their common best learning rate (3×10^{-3}). Headroom appears to grow with depth at 30 000 steps, but only because `ssm_only` degrades; both arms are past overfitting there, and at each arm's best configuration (four blocks, 6000 steps) the difference is not reliably signed across seeds.

	6000 steps		30 000 steps	
Blocks	`ssm_only`	`dense`	`ssm_only`	`dense`
1	13.65	13.69	13.46	13.28
2	13.17	13.43	14.00	13.30
4	**12.32**	12.84	14.41	13.58

Table 5 gives the full grid behind Section 9's depth control. Both arms use their common best learning rate. The apparent growth of headroom with depth at 30 000 steps is an overfitting artifact:

`ssm_only` degrades while `dense` stays flat, and every $30\,000$-step run is worse than its 6000-step counterpart.

P METHODOLOGICAL FINDINGS

Two experimental-design artifacts were large enough to invert conclusions, and we report them because both are invisible in a results table.

Blending form. Whether attention *replaces* or is *added to* the recurrent path swings measured accuracy from 0.008 to 0.983 with nothing else changed. The replacing form disables the SSM at routed positions and does not correspond to what deployed hybrids do.

Eval-time mask ablation. Swapping the routing mask after training systematically answers a different question than the one intended, and led us to a conclusion we later retracted.

Variance tracks attention activity. On real text, seed variance rises monotonically with the number of attention-active positions, from 0.8% for a sparse oracle to 4.8% for dense attention. Single-seed comparisons between arms with different attention density are therefore not safe.

Q A RETRACTED RESULT, AND WHAT CAUSED IT

An earlier version of this work reported a $\sim 90\times$ separation between static and heuristic allocation at high positional entropy. That result does not survive correcting a single architectural choice, and we document it because the artifact is not specific to our setting.

The original runs blended attention and the recurrent path by *switching*: $m \cdot \text{attn} + (1 - m) \cdot \text{ssm}$, which disables the SSM at routed positions. Deployed hybrids add rather than switch. Re-running the identical experiment with the additive form:

Arm (MQAR, $H = 4.53$)	switch (as reported)	additive (correct)
static	0.008	0.983
heuristic	0.727	0.987
oracle	0.997	0.998

All three arms fall within 1.5 points. The separation the earlier claim rested on was created by the blending form, not by the allocation policy. A second artifact compounded it: evaluating by swapping the routing mask *after* training answers a different question than training under the mask, and had supported the same erroneous conclusion.

We report both because a reader comparing allocation policies in a hybrid model can reproduce either mistake without noticing, and because the effect sizes involved (0.008 versus 0.983) exceed anything else measured in this paper.