RECURRENT REFINEMENT MOVES CANDIDATE-QUALITY READABILITY EARLIER AND STABILIZES ITS COORDINATES

Anonymous authors
Paper under double-blind review

ABSTRACT

A looped language model applies the same layer stack several times to a token position, so "physical layer 8" is not one event but one per pass. This makes a question well-posed that a feed-forward stack cannot ask: *when* during recurrent refinement does a property of the computation become linearly readable at a given layer, and is it carried by the same coordinates throughout? We answer it for candidate-quality information in a frozen 2.6B looped transformer (Ouro), with small bias-free antisymmetric taps selected from a fixed domain-subset sweep and evaluated on a task-disjoint five-domain candidate-ranking target under an audited pairwise protocol. Two evaluation modes separate two questions. *Locus-local refit* asks whether the information is readable at a loop/layer cell: held-out macro top-1 at physical layers 8 and 16 is weak on the first pass (0.410, 0.440; chance 0.282) and statistically tied with the established late-layer basis by loops 3–4 (0.633, 0.628), with loop 4 − loop 1 contrasts of $+0.19$ to $+0.21$ at every tested layer. *Frozen cross-loop transplant* asks whether it is carried by functionally stable readout coordinates: taps fitted at loop 1 lose 0.09–0.23 when applied unchanged at loop 4, while taps fitted at loops 2–3 transfer at parity ($|\Delta| \leq 0.024$). This indicates a first-pass coordinate change followed by stable frozen-transfer behavior. Both signatures replicate on three further Ouro checkpoints, and the archived RLTT taps transfer frozen across checkpoints within 0.028 of locally refit taps; the depth trend replicates qualitatively in an out-of-family depth-recurrent model (Huginn, $+0.107$ from step 1 to step 8), where no comparable immediate first-pass transfer discontinuity is detected. A non-looped control shows that a same-class readout exists without recurrence, so we claim a property of *where and when* the signal is readable in looped models, not that looping creates it. All results are readout-only; no intervention is performed.

1 INTRODUCTION

A standard transformer processes each token position through its layer stack once. Looped (universal) transformers instead apply a fixed stack repeatedly, carrying the hidden state forward between passes, so that a single position is refined by the same weights several times (Dehghani et al., 2019; Geiping et al., 2025; Zhu et al., 2025; Saunshi et al., 2025). This changes what can be asked of a probe. In a feed-forward model, a linear readout at layer ℓ answers one question: is the property readable at depth ℓ? In a looped model the same layer is visited on every pass, and two further questions appear that have no feed-forward analogue: *when* during refinement a property becomes readable at a given physical layer, and whether the same coordinates carry it from one pass to the next.

This paper answers both questions for a specific, well-audited property: the relative quality of candidate continuations. The prior readout work on this model family fitted its taps at a fixed late-layer basis (physical layers 24, 36, and 47) selected from *final-pass* geometry (Kirin, 2026b). That basis records where the signal is readable after the model has finished refining, and says nothing about when during the refinement it becomes readable, or whether the same coordinates carry it throughout. Because Ouro applies the *same* forty-eight layers on each of four passes, these are separable questions — physical layer 8 at loop 3 is the same weights as physical layer 8 at loop 1, applied

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.

to a state already refined twice — and they are only well-posed in a looped model, since in a feed-forward stack "layer 8" happens once.

Two modes, two questions. Our central methodological point is that a single probe-accuracy matrix over loops and layers conflates the two questions, and that they are answered by different evaluation modes (§3). A *locus-local refit* trains and selects a fresh tap at every loop/layer cell and reports the best a matched low-capacity reader can do there; it answers readability. A *frozen cross-loop transplant* takes the tap selected at an earlier pass and applies it unchanged — same weights, no bias, no normalization, nothing to re-fit — at a later pass of the same physical layer; it answers whether the same readout coordinates remain functionally valid. The two can disagree, and they do: the first-pass cells are both weakly readable *and* poorly served by the coordinates that work on later passes.

Findings. On the frozen Ouro-RLTT 2.6B checkpoint, under a task-disjoint split and an antisymmetrized pairwise protocol summarized in Appendix E, we find (§4):

- **Readability rises across passes at every tested layer.** Held-out macro top-1 at physical layers 8, 16, and 24 rises by $+0.19$ to $+0.21$ from loop 1 to loop 4 (task-clustered 95% intervals exclude zero at every layer). By loop 3 the early layers are statistically tied with the late-layer basis; four cells (layers 8 and 16 at loops 3 and 4) satisfy the paper's parity criterion.
- **Frozen transfer indicates a first-pass coordinate change and later functional stability.** All nine taps sourced at loop 1 lose 0.085–0.233 when transplanted frozen to a later loop; all nine taps sourced at loops 2–3 transfer within ± 0.024 with score correlations of 0.90–0.99.
- **Both signatures are properties of the family, not of one checkpoint** (§5). The same fixed protocol replicates on the base, Thinking, and 1.4B Ouro checkpoints, and the RLTT-fitted taps themselves transfer frozen to the base and Thinking feature spaces within 0.028 of locally refit taps. On Huginn, an out-of-family depth-recurrent model, the depth trend replicates qualitatively ($+0.107$ from step 1 to step 8) but no comparable immediate first-pass transfer discontinuity is detected.
- **Looping is not what makes the signal readable** (§6). A conventional non-looped transformer under the identical protocol yields a same-class readout (0.568 macro top-1 vs. 0.631). We therefore claim a result about *where and when* candidate quality is readable in a looped model, and explicitly not that recurrence creates the readout.

What this paper is not. Every result is a readout. No intervention is performed at any locus, and we do not claim that the early, stable cells are steerable. The frozen-transfer result constrains one reading of looped models — that shared weights are semantically stationary across passes — and we state the mechanistic interpretation we find most natural (the first pass uses a distinct coordinate system for related information) as an interpretation, not as a tested mechanism.

2 Setup

2.1 The model

Ouro-RLTT is a 2.6B-parameter looped (universal) transformer (Zhu et al., 2025; Williams & Tureci, 2026): a fixed stack of forty-eight transformer layers applied over four *loop iterations*, so that a single token position is processed by the same weights four times, with the hidden state carried forward between iterations. Reasoning-oriented training (the RLTT variant) encourages the model to use these iterations to refine intermediate computation before committing to output. We use the model entirely **frozen**: no backbone weight is updated anywhere in this paper. Every readout in this work is an external probe trained on top of frozen activations.

The relevant consequence of the looped design is that each generation step yields not a single hidden vector per layer but a small **trajectory**: the same position revisited across four loop iterations produces four successive hidden states per layer, and it is this trajectory — the model's own iterative refinement of its computation — that we read.

We write $h^{(\ell,u)} \in \mathbb{R}^{2048}$ for the mask-valid mean-pooled hidden state of a candidate at physical layer ℓ on loop $u \in \{1, \ldots, 4\}$, and call the pair (ℓ, u) a *cell*, abbreviated ℓ_Lu (so 8_L3 is layer 8 on the third pass). For two candidates A, B we write $\Delta h = h_A - h_B$.

2.2 TAPS

Each readout is produced by a small head trained on top of the frozen trajectory; the backbone is never updated. That such small heads suffice is itself part of the claim: the information is present in the hidden geometry in a form a low-capacity reader can extract, rather than requiring a large external model to manufacture.

The move from a large evaluator to taps was both scientific and methodological. The GRU-over-loops evaluator of Kirin (2026a) was large enough to reveal that useful structure existed in the loop trajectory, but the same capacity also made it able to exploit presentation-order regularities under fixed-order training (Kirin, 2026b). The tap design deliberately moves in the opposite direction: small heads, explicit pairwise differences, bias-free scoring, and swap-safe antisymmetry by construction. This makes the readouts less expressive but easier to audit. A tap that succeeds under these constraints is stronger evidence that the information is present in the hidden-state geometry rather than manufactured by the external evaluator.

The tap used throughout this paper is `AntisymLinearNoNorm`, which forms the pairwise difference $\Delta h = h_A - h_B$ and applies a bias-free linear readout, $s_{\text{raw}}(A, B) = w^\top (h_A - h_B)$, so that swapping candidates flips the score sign: $s(B, A) = -s(A, B)$ by construction. It preserves raw magnitude, unlike the layer-normalized variant $s_{\text{LN}}(A, B) = w^\top \text{LN}(h_A - h_B)$ used by the production CoreContent selector of Kirin (2026b). The archived run artifacts confirm that the 0.6467/0.6200 L4_36/L4_47 references are same-run, same-subset local refits with this NoNorm tap; the public-family replications independently refit the same tap class, while Huginn applies the same trainer on recurrence steps 1–8. Optimizer and refit-selection details are given in Appendix A.

2.3 TARGET AND PROTOCOL

The target is *CoreContent v2*: candidate-quality ranking over constructed candidate groups in five content domains (alignment, reasoning, math, coding, logic), where each group contains alternatives tied to one source task and relative-quality labels come from the source preference data or external answer keys/verifiers. The tap must rank the designated-best candidate first. Under a strictly task-disjoint split (zero task IDs crossing the train/held-out boundary), the late-basis tap of Kirin (2026b) reaches held-out macro top-1 **0.6310**, against a broad-objective baseline of **0.5525**; this corrected a previously reported 0.6691 measured on a split in which 195 task IDs crossed the boundary. That history is why every split in this paper is asserted task-disjoint in code.

The evaluation protocol is source-item-disjoint and antisymmetrized throughout; Appendix E states the relevant integrity rules self-containedly for review. The data are a bounded deterministic subset of the CoreContent v2 set on the existing task/prompt-disjoint split (250 train / 60 validation / 120 held-out groups per domain; 8,592 candidates; **zero** task IDs crossing, asserted). The tap is the bias-free `AntisymLinearNoNorm` head, trained by the fixed pairwise procedure, with its 36-run selection sweep (16 distinct optimization trajectories; Appendix A) chosen on validation macro top-1 only. At eleven of the twelve transfer-bearing cells, validation selects the `code_reasoning` training subset (coding+reasoning; 1,714 pair differences), while held-out evaluation remains macro-averaged over all five domains; Appendix B reports the selected configurations. Intervals are task-clustered bootstraps (10,000 draws, clusters = task ID within domain), with identical draws applied to both members of every paired comparison; matched chance is the analytic tie-aware expectation, macro **0.2818**.

2.4 THE LATE-LAYER BASIS, AND WHY THIS STUDY EXTENDS IT DOWNWARD

The canonical readout basis for this model family — physical layers **24, 36, and 47** across the four loops — was selected by a prior locus programme (Kirin, 2026a;b) from pairwise locus and loop ablations, normalization and bias-decomposition passes, and all-layer cached probes, before any of the results reported here; it was not chosen from an architectural prior or from the present headline

numbers. Operationally, layer 24 acts as a mid-depth representation before final consolidation, layer 36 as a late integration point, and layer 47 as the terminal/pre-output boundary.

We therefore extended the locus study downward, to physical layers **8** and **16** across all four loops, with layer 24 measured in the same run as an internal control and the historical L4_36 / L4_47 readouts as references.

Features are one frozen forward per candidate (`use_cache=False`, four UT steps, no early exit) at layers $\{8, 16, 24, 36, 47\}$ × loops {L1–L4}, mask-valid mean pooled, stored $[5, 4, 2048]$ fp16.

3 Two evaluation modes for two questions

Two evaluation modes answer two different questions, and conflating them is the main way this measurement can mislead.

Locus-local refit asks *is the information readable here?* A fresh tap is trained and selected at each loop/layer cell independently, so each cell reports the best that a matched low-capacity reader can do at that locus.

Frozen cross-loop transplant asks *is it carried by the same coordinates?* The tap selected at source loop u is applied **unchanged** — same weights, same feature convention, and by construction no bias, normalization, or threshold to re-fit — at a later loop $v > u$ of the same physical layer.

The two modes are logically independent. A cell can be readable under local refit while the coordinates that read it fail on the next pass; a cell can also be poorly readable while whatever weak readout exists is preserved downstream. Reporting only the local-refit matrix — the usual probing practice — silently assumes that accuracy at a cell says something about continuity across passes. It does not. The transplant mode is what makes the statement "the same coordinates remain functionally valid from pass 2 onward" an empirical claim rather than an interpretation of an accuracy table. Because the tap has no bias, no normalization, and no threshold, a transplanted tap cannot be silently re-calibrated at the target; any change in score is a change in how the target representation projects onto the frozen source readout.

Each transplanted tap was required to reproduce its stored source-cell score exactly before target evaluation, and its weight hash was asserted unchanged across evaluation, so any loss is a property of the representation and not of the transfer machinery.

Throughout, we use *rotation* and *coordinate stability* operationally: a source tap that loses against the target-local refit is evidence that its readout coordinates are no longer functionally equivalent at the target, whereas parity together with high score correlation is evidence of functional stability. We did **not** compute cosine angles or another direct weight-space rotation statistic between independently refit taps.

Parity criterion. A cell is labelled *at parity* with the late-layer basis if it is above matched chance *and* within 0.03 macro top-1 of the final-loop layer-24 refit (24_L4). This is the `EARLY_READABLE` criterion used in the archived OPI analysis. A contemporaneous run plan and preserved analysis code place the criterion, the 0.03 margin, and the pooling convention before feature extraction and before result generation; the plan was not git-sealed or externally preregistered, so we describe these choices as *pre-specified*, not preregistered.

4 Results on Ouro-RLTT

4.1 Readability rises across the loops, at every layer

Held-out macro top-1, with task-clustered 95% intervals; every cell excludes matched chance (Table 1, Figure 1A).

Table 1: Locus-local refit. Held-out macro top-1 with task-clustered 95% intervals, by physical layer and loop. Chance is 0.2818. Bold cells satisfy the parity criterion of §3. References at the same protocol: L4_36 **0.6467** [.610, .683], L4_47 **0.6200** [.583, .657].

Physical layer	Loop 1	Loop 2	Loop 3	Loop 4
8	0.4100 [.373, .447]	0.5683 [.532, .603]	**0.6333** [.597, .670]	**0.6217** [.585, .658]
16	0.4400 [.403, .477]	0.5833 [.547, .620]	**0.6167** [.580, .653]	**0.6283** [.592, .665]
24	0.4283 [.392, .463]	0.5683 [.532, .605]	0.6100 [.573, .647]	0.6183 [.580, .655]

The loop trend is large and significant at every layer: L4 − L1 is **+0.212** [+0.163, +0.260] at layer 8, **+0.188** [+0.145, +0.235] at layer 16, and **+0.190** [+0.147, +0.235] at layer 24. By loop 3, the early layers have caught up with the established basis: the paired difference between 8_L3 and the final-loop layer-24 refit is +0.015 [−0.022, +0.050], and against the strongest historical reference L4_36 it is −0.013 [−0.047, +0.020] — statistical ties in both cases. Layers 8 and 16 at loops 3 and 4 are the four cells that satisfy the parity criterion (above chance *and* within 0.03 of the layer-24 final-loop refit). Loop 2 does not: 8_L2 and 16_L2 are far above chance but still 0.035–0.050 short, and we do not describe them as at parity.

4.2 Frozen transfer is stable from loop 2 onward and changes sharply at L1→L2

All nine transplants sourced at loop 2 or 3 retain the target-local performance (Δ within ± 0.024; Spearman correlation with the target-local refit 0.90–0.99; pairwise sign agreement 0.92–0.97), and none of their intervals excludes zero. All nine transplants sourced at loop 1 lose materially (0.085–0.233; Spearman as low as 0.53 at layer 8), and every one of those deltas excludes zero (Table 2, Figure 1B).

Table 2: Frozen cross-loop transplant into loop 4. Target macro top-1, with the paired difference against the target-local refit in parentheses and Spearman score correlation ρ on identical held-out candidates. The full 18-transfer matrix is in Appendix B.

Physical layer	L1 → L4	L2 → L4	L3 → L4
8	0.4100 (−0.212), $\rho = 0.53$	0.6283 (+0.007), $\rho = 0.94$	0.6283 (+0.007), $\rho = 0.91$
16	0.4967 (−0.132), $\rho = 0.78$	0.6283 (0.000), $\rho = 0.95$	0.6300 (+0.002), $\rho = 0.99$
24	0.5100 (−0.108), $\rho = 0.77$	0.6100 (−0.008), $\rho = 0.90$	0.6367 (+0.018), $\rho = 0.95$

All three headline L1→L4 transfers are exact-configuration matched; Appendix B gives the full selection-configuration audit. Eleven of the twelve transfer-bearing cells select `code_reasoning`. The resulting readout still ranks held-out groups across all five domains, including math, logic, and alignment, but because the full five-domain validation set participates in selection we treat this as cross-domain transfer of the selected readout rather than domain-OOD generalization.

4.3 Controls

A label-shuffle rerun of the full grid at four representative cells gives 0.350 (8_L1), 0.4267 (8_L4), 0.4033 (16_L2) and 0.3367 (24_L4). The shuffle floor sits well above naive chance — the alignment domain contributes a two-candidate 0.5 pairwise floor, and validation selection over the fit sweep adds optimism — so the informative comparison is each cell against *its own* shuffle counterpart. On that comparison the weak first-pass cell 8_L1 clears its shuffle by only +0.060, while 8_L4 clears by +0.195, 16_L2 by +0.180, and 24_L4 by +0.282. First-pass readability at layer 8 is therefore weak in absolute terms and barely distinguishable from its own shuffled control; the matched shuffle was run at four representative cells only, so we do not extend that second, stronger statement to 16_L1, which was not shuffle-tested. Ranking candidates by text length gives macro 0.323, so the taps are not length proxies. The new extraction path reproduces the production cache on 2,434 identical held-out candidates, and layers 8 and 16 are confirmed to be genuinely new loci rather than re-labelled cached ones. Twenty targeted tests pass. Full control detail is in Appendix C.

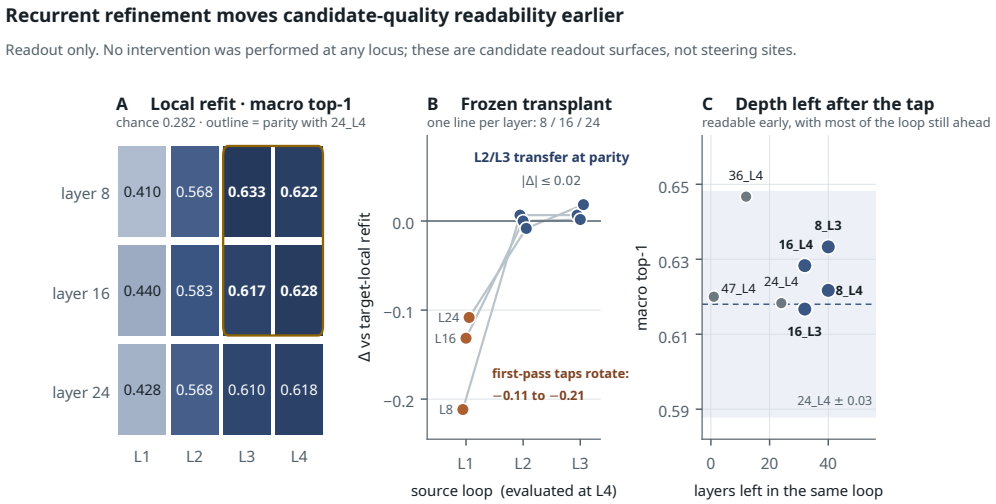

Figure 1: Recurrent refinement moves candidate-quality readability earlier in the reused physical stack. (A) Locus-local refit across loops and layers, with the historical L4 references; chance is 0.282. (B) Frozen cross-loop transplant, as the paired loss against the target-local refit: taps sourced at loop 1 lose 0.09–0.23, taps sourced at loops 2–3 lose nothing measurable. (C) The four early cells that reach parity still leave 32–40 layers of same-loop downstream depth after the tap point. All values are task-clustered; the L1→L2 transfer loss is the single largest coordinate discontinuity in the matrix. "Rotation" denotes this operational frozen-transfer effect, not a directly measured tap-weight angle.

4.4 What this establishes, and what it does not

It establishes that process-quality information is linearly readable at physical layers 8 and 16 during loops 3 and 4 at parity with the historical mid/late basis; that readability increases sharply from the first loop to the later ones at every tested layer, with the L3 and L4 cells statistically indistinguishable from one another; and that frozen readout coordinates are functionally stable across loops once at least one full pass has completed, with the largest transfer discontinuity concentrated at the L1→L2 transition. The natural mechanistic reading is that the shared weights are semantically stationary for this readout only from the second pass onward: the first pass appears to use a distinct coordinate system for related information. It does **not** establish that a single universal quality scalar is being moved forward, that these loci are steerable, or that looping is necessary for candidate-quality readability — the non-looped control of §6 speaks directly against the last of these. No intervention was performed here. We label the four qualifying cells *candidate readout surfaces*: they are readable, shallow, and leave 40 and 32 layers of same-loop downstream depth respectively, which is exactly the profile an intervention study would want — and is not itself evidence that an intervention would work.

5 Replication within the family and out of it

5.1 Within-family replication and frozen transfer

Everything above concerns one checkpoint. We re-ran the same fixed protocol on the base 2.6B, 2.6B-Thinking, and 1.4B Ouro checkpoints (Figure 2A). Every early cell is weak on the first pass (0.408–0.473) and reaches 0.60–0.65 by loops 3–4, with every loop contrast excluding zero. At the earliest captured layer, L1→L4 loses 0.178–0.250 across the three replications while later-source transfers are much smaller (Figure 2C). Because local refits are independently selected over training subset and learning rate, Appendix F audits those configurations and we treat the transfer replication as supporting evidence; the exact-configuration headline coordinate result is RLTT. Independently, the archived RLTT taps transfer **frozen across checkpoints** to the base and Thinking feature spaces within 0.028 macro top-1 of local refits, with median Pearson/Spearman score correlations

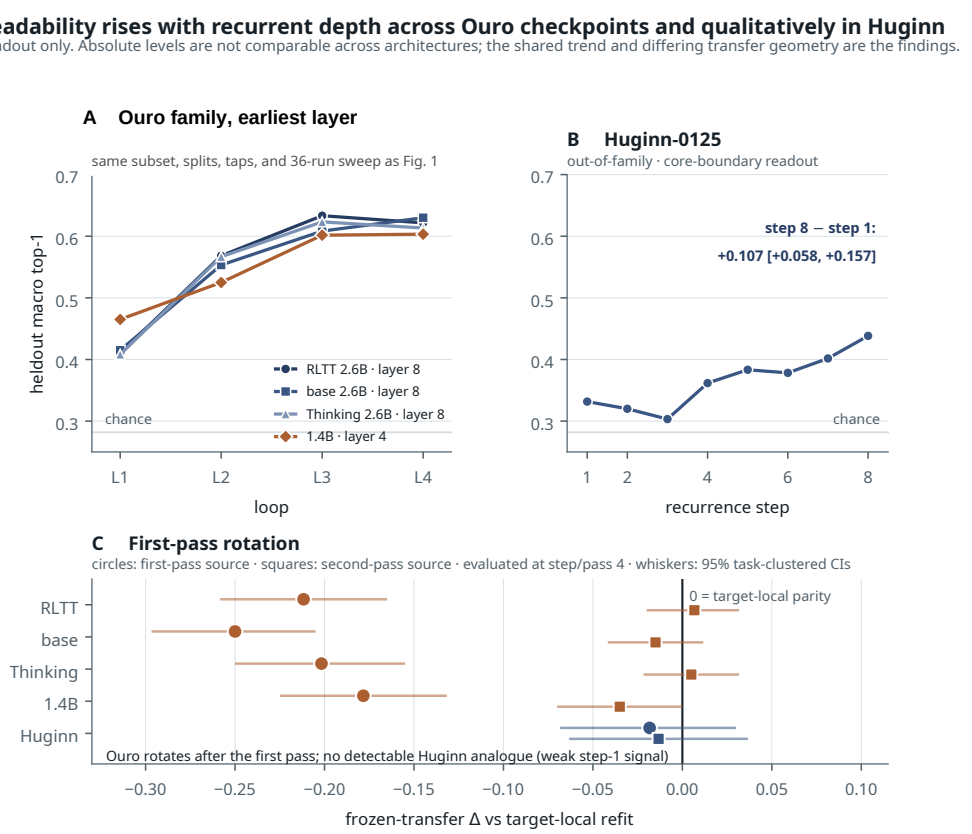

Figure 2: The recurrent-depth readability trend beyond RLTT. (A) Local-refit macro top-1 at the earliest captured layer across loops, for all four Ouro checkpoints (1.4B at its mapped layer 4); the identical fixed subset, splits, and training procedure as Figure 1. (B) Huginn-0125, an out-of-family depth-recurrent architecture, at its recurrent-core boundary across steps 1–8: the trend replicates qualitatively (step 8 − step 1 = +0.107 [+0.058, +0.157]) at a lower absolute level. (C) Frozen-transfer Δ against the target-local refit, first- and second-pass-sourced taps evaluated at pass/step 4; whiskers are 95% task-clustered bootstrap CIs. Every Ouro checkpoint shows a significant first-pass loss at the earliest captured layer. Huginn shows no comparable immediate discontinuity at step 4: L1→L2–4 intervals include zero; losses emerge only later (L1→L5, L1→L8, and L2→L8 exclude zero), while step-4-sourced taps remain within 0.027 of later local refits.

of 0.975/0.973 on Base and 0.996/0.995 on Thinking. On the 1.4B, early-loop cells can exceed its late references (e.g. 12_L3 0.623 vs. 18_L4 0.568 and 23_L4 0.560), so the "late basis is strongest" framing is specific to 2.6B.

5.2 Out-of-family replication on Huginn

On Huginn-0125 (Geiping et al., 2025), an out-of-family 3.5B depth-recurrent transformer, the recurrent-core readout rises from 0.332 at step 1 to 0.438 at step 8 (chance 0.282; contrast +0.107 [+0.058, +0.157]). Transfer dynamics differ from Ouro: exact-configuration L1→L4 is only −0.018 (interval includes zero), while losses appear later at L1→L5 (−0.060), L1→L8 (−0.093), and L2→L8 (−0.090); step-4-sourced taps stay within 0.027 of later local refits. Thus the depth trend replicates, but the immediate first-pass discontinuity does not; because selected training subsets also change across Huginn steps, its transfer matrix is descriptive rather than evidence for cross-architecture coordinate stability.

5.3 What generalizes and what does not

Multi-checkpoint support is therefore strongest for three bounded statements: readability rises with recurrence depth (all four Ouro checkpoints and qualitatively Huginn); Ouro shows a repeatable first-pass-versus-later transfer asymmetry, with RLTT providing the exact-configuration headline evidence; and frozen cross-checkpoint transfer shows shared readable geometry within the 2.6B Ouro family. Huginn lacks the immediate L1 discontinuity and instead develops later drift. Earlier-in-physical-depth migration remains specific to Ouro's reused-stack design, and the late basis being strongest is specific to 2.6B. Absolute readability levels are not compared across architectures.

6 Is the loop necessary? A non-looped control

The readouts above are all measured on a looped model, which leaves open a question the paper must answer before attributing anything to recurrence: **is a looped architecture necessary for quality-relevant information to be linearly readable at all?** We test this directly by repeating the CoreContent-v2 protocol on a conventional, non-looped transformer.

The control backbone is MiniCPM-2B-sft-bf16 (Hu et al., 2024): a standard decoder transformer with **40 physically distinct blocks** (no recurrent block reuse), hidden size 2,304, explicitly SFT-trained. We extract frozen features at two physical layers (24 and 36) with mask-valid mean pooling, run the identical CoreContent probe grid, and evaluate under a deterministic **task-disjoint** split (zero task IDs crossing the boundary).

Table 3: Non-looped control under the identical task-disjoint CoreContent-v2 protocol.

Backbone	Held-out macro top-1	Held-out macro pairwise
Ouro-RLTT (looped), corrected task-disjoint	**0.6310**	—
MiniCPM-2B-sft (non-looped), task-disjoint	**0.5680**	0.7237

The Ouro value 0.6310 in Table 3 is the corrected late-basis CoreContent-v2 result on CoreContent's full five-domain task-disjoint evaluation set, carried over because the MiniCPM control repeats that full CoreContent-v2 protocol. It is therefore not a single-cell value from the 2,150-group NoNorm localization subset of Table 1; for example, 24_L4 is 0.6183 and the same-subset 36_L4 reference is 0.6467.

What this establishes. A conventional non-looped SFT transformer contains linearly readable candidate-quality information that generalizes across strictly task-disjoint data: **a looped architecture is not necessary for this class of readout to exist.** The 6.3-point gap between Ouro (0.631) and the control (0.568) is *not* an architecture comparison; the models differ in family, data, objective, width, parameter budget, and tap geometry, so we do not attribute the gap to looping.

What remains loop-specific is the pair of questions in §3: when a fixed physical layer becomes readable, and whether the same coordinates carry the signal across repeated applications of the same weights. A non-looped model has no recurrence axis and therefore cannot confirm or refute either the depth trend or the first-pass rotation. A causal architecture comparison would require matched looped and non-looped models trained identically, which we have not run.

7 Related work

Looped and recurrent language models. Universal and recurrent-depth transformers reuse parameters across depth or recurrent steps (Dehghani et al., 2019; Zhu et al., 2025; Williams & Tureci, 2026; Geiping et al., 2025; Saunshi et al., 2025; Bae et al., 2025). Most work asks what extra recurrent depth buys in capability; we instead study the *geometry of one fixed readout across repeated applications of the same weights*.

Readability and dynamics across loops. Chen et al. (2026) vary Ouro's configured loop count from 1 to 8 and report slightly declining late-locus linear-probe accuracy as total recurrence increases, alongside a separate final-loop concept-injection effect. Our axis is different: within a fixed

four-pass trajectory we resolve physical layer × pass and frozen transplant, so their late-locus degradation and our earlier-in-physical-depth migration can coexist. Blayney et al. (2026) instead analyze whole-state and attention dynamics: their §4.1/Fig. 4 show approximate fixed-point convergence for Huginn and retrofitted Llama but not Ouro. Our claim is target-specific—one candidate-quality readout becomes functionally stable from loop 2 onward—and neither requires nor implies convergence of the full residual state.

Readout visibility versus recurrent state. Sharma & Vu (2026) show that dense per-loop supervision can make intermediate exits usable while normalized readouts remain blind to an active recurrent variable, hidden-state scale. Their target is training-time visibility/control of radial scale; ours is post-hoc candidate-quality separability. The common limitation is that readout success establishes accessibility only in the coordinates exposed by that readout.

Layer-wise readability and probing. Logit/tuned-lens methods decode intermediate feed-forward states (nostalgebraist, 2020; Belrose et al., 2023), while linear probes measure task-relevant separability across depth (Alain & Bengio, 2016). Recurrence separates depth from layer identity: local refit measures readability at each cell, and frozen transplant asks whether the same coordinates remain readable later. Matched label-shuffle controls follow Hewitt & Liang (2019).

Relation to the prior programme. The taps, CoreContent target, late-layer basis, and correction history come from a broader programme on frozen Ouro-RLTT (Kirin, 2026a;b); Appendix E states the evaluation-integrity rules needed here. This paper isolates recurrent readout dynamics and relies on none of that programme's intervention claims.

8 LIMITATIONS

The recurrent-depth trend is family-replicated, not universal. It holds across four Ouro checkpoints and, as a depth trend only, on Huginn; the earlier-in-physical-depth formulation remains specific to Ouro's reused-stack design, and Huginn shows no comparable first-pass frozen-transfer discontinuity.

The early-layer result is readout-only and target-bounded. We perform no intervention and claim no controllability or capability gain. The result uses one target (CoreContent candidate quality), one bounded subset (120 held-out groups per domain), and one pooling convention; 195 `hendrycks_math` task IDs crossing the historical split were excluded. At eleven of the twelve transfer-bearing cells the validation-selected tap is fit only on coding+reasoning pairs, although selection and held-out evaluation span all five domains; this is therefore not a domain-OOD test. The frozen CoreContent direction also fails to transfer to generated-branch correctness (AUROC 0.354–0.475; Appendix D), so this is not a universal quality axis.

The readout does not require recurrence. A non-looped SFT transformer reaches 0.5680 under the task-disjoint protocol (§6). The Ouro–control gap is not architecture-controlled; attributing it causally would require matched looped/non-looped models trained identically.

What would change the conclusion. Failure of the loop trend on a second readout target, or pooling-specific first-pass transfer behavior, would weaken the claim; both-signature replication in another looped family and a matched architecture study would strengthen it.

USE OF AI TOOLS

The author designed the experiments, ran the experimental code, and produced all results reported here. Two AI assistants (Claude, Anthropic; ChatGPT 5.6 Sol, OpenAI) were used as adversarial reviewers and discussion partners on the analysis and framing, to search and summarize related literature, and to draft and edit prose from the author's notes and results. Claude was additionally used for implementation and debugging in parts of the later Ouro-family and Huginn replication pipeline, including edits after a failed cross-model attempt; surviving records do not establish the same for the original localization run. All AI-drafted text was revised by the author; all AI-suggested references were checked against the primary sources; all numerical claims were verified by the author against the stored experimental artifacts. The author takes responsibility for the final content.

REPRODUCIBILITY STATEMENT

All results are readouts on frozen checkpoints. The public backbones are ByteDance/Ouro-2.6B (pinned 1ed0425), Ouro-2.6B-Thinking (pinned f1edd81), Ouro-1.4B, tomg-group-umd/huginn-0125 (bf16), and MiniCPM-2B-sft-bf16. The Ouro-RLTT checkpoint of Williams & Tureci (2026) is not available from a public model repository. For review-time reproducibility, the exact checkpoint used in this work is available through an anonymous Zenodo review artifact. No post-review availability or general redistribution license is asserted. The complete RLTT analysis package is nevertheless preserved locally: the checkpoint, all fourteen fitted `AntisymLinearNoNorm` cell readouts, eighteen feature shards, the 2,150-group zero-crossing split, all frozen cross-loop transplant evaluations, 10,000-draw bootstrap outputs, logs, environment records, manifests, and checksums. The checkpoint safetensor hashes still match the original critical-model manifest, the stored model-configuration hash matches the run record, all per-cell tap-weight hashes recomputed from the retained bundle match the recorded results, and the original run directory verifies 100/100 entries against its `SHA256SUMS`. The public-model replication packages are also preserved and checksum-verified (176/176 family files and 63/63 Huginn files), including local refits, matrices, bootstrap statistics, revision pins, and the frozen RLTT-to-Base/Thinking cross-model evaluations. The two Ouro signatures replicate under the same fitting protocol on the public base 2.6B, Thinking 2.6B, and 1.4B checkpoints; moreover, the archived RLTT taps transfer directly to the two public 2.6B siblings within 0.028 macro top-1 of their local refits, with median Pearson/Spearman score correlations of 0.975/0.973 on Base and 0.996/0.995 on Thinking (§5; Appendix F). Thus the headline backbone cannot be re-fetched from a public model repository, but the exact checkpoint is available during review through that artifact, the reported analysis remains locally hash-auditable, and the family-level signatures are independently runnable on public weights. For the original localization run, the pooling convention and `EARLY_READABLE`/0.03 rule were fixed in a contemporaneous plan and code before feature extraction/results, but the plan was not git-sealed or externally preregistered; the later Huginn plan predates extraction, whereas the Ouro-family plan was recorded only after base-2.6B extraction had begun. Appendix C lists the twenty targeted tests that the pipeline passes. For review, an anonymized supplement provides the retained redistributable analysis artifacts, predictions, refit outputs, split/integrity manifests, statistics, environment records, public-checkpoint replication material, and verification files needed to audit the reported analyses. That supplement does not embed the multi-gigabyte RLTT weights; checkpoint access during review is through the anonymous Zenodo review artifact linked above.

REFERENCES

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

Amos Azaria and Tom Mitchell. The internal state of an LLM knows when it's lying. In *Findings of EMNLP*, 2023.

Sangmin Bae, Adam Fisch, Hrayr Harutyunyan, Ziwei Ji, Seungyeon Kim, and Tal Schuster. Relaxed recursive transformers: Effective parameter sharing with layer-wise LoRA. In *International Conference on Learning Representations*, 2025.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*, 2023.

Hugh Blayney, Álvaro Arroyo, Johan Obando-Ceron, Pablo Samuel Castro, Aaron Courville, Michael M. Bronstein, and Xiaowen Dong. A mechanistic analysis of looped reasoning language models. *arXiv preprint arXiv:2604.11791*, 2026.

Guanxu Chen, Dongrui Liu, and Jing Shao. Loop as a bridge: Can looped transformers truly link representation space and natural language outputs? *arXiv preprint arXiv:2601.10242*, 2026.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal transformers. In *International Conference on Learning Representations*, 2019.

Jonas Geiping, Sean McLeish, Neel Jain, et al. Scaling up test-time compute with latent reasoning: A recurrent depth approach. *arXiv preprint arXiv:2502.05171*, 2025.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of EMNLP-IJCNLP*, 2019.

Shengding Hu, Yuge Tu, Xu Han, et al. MiniCPM: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*, 2024.

Saurav Kadavath, Tom Conerly, Amanda Askell, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Jan Kirin. Relational preference encoding in looped transformer internal states. *arXiv preprint arXiv:2604.09870*, 2026a.

Jan Kirin. Operational proto-introspection in looped language models: Process-quality taps, executable branching, and the readout–control boundary. *arXiv preprint arXiv:2607.18553*, 2026b.

Hunter Lightman, Vineet Kosaraju, Yura Burda, et al. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

nostalgebraist. Interpreting GPT: the logit lens. LessWrong / AI Alignment Forum post, 2020. URL `https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-`.

Nikunj Saunshi, Nishanth Dikkala, Zhiyuan Li, Sanjiv Kumar, and Sashank J. Reddi. Reasoning with latent thoughts: On the power of looped transformers. *arXiv preprint arXiv:2502.17416*, 2025.

Rituraj Sharma and Tu Vu. Dense supervision is not enough: The readout blind spot in looped language models. *arXiv preprint arXiv:2606.24898*, 2026.

Jonathan Williams and Esin Tureci. Prioritize the process, not just the outcome: Rewarding latent thought trajectories improves reasoning in looped language models. *arXiv preprint arXiv:2602.10520*, 2026.

Rui-Jie Zhu, Zixuan Wang, Kai Hua, et al. Scaling latent reasoning via looped language models. *arXiv preprint arXiv:2510.25741*, 2025.

A TAP ARCHITECTURE AND TRAINING DETAILS

We use the term *tap* for small readout heads trained on frozen hidden-state features. This is deliberately narrower than the original evaluator. The evaluator of Kirin (2026a) used token attention pooling for each loop state, normalized candidate differences, projected them to a 512-dimensional hidden space, ran a two-layer unidirectional GRU over the four loop states, concatenated the GRU output with the final-loop projection, and passed the result through a nonlinear scorer. That architecture was useful as a discovery tool because it could integrate information across the loop trajectory, but subsequent locus and swap-control work showed that the GRU was repeatedly weak or mildly counterproductive relative to simpler exact-antisymmetric heads.

Both `AntisymLinear` and `AntisymLinearNoNorm` are structurally antisymmetric: $s(B, A) = -s(A, B)$ up to numerical tolerance and the exact details of the symmetric preprocessing. The difference is interpretive. `AntisymLinear` suppresses raw scale and norm effects and asks whether the *direction* of the pairwise difference carries the label; it is the safer comparator for hard near-miss distinctions where magnitude can be a shortcut. `AntisymLinearNoNorm` preserves raw magnitude and often behaves more like a scalar utility readout; it is useful when quality is relatively transitive or when easy-prune/objective distinctions are carried by feature scale.

For the localization refits, the archived training configuration expands to 36 runs: six fixed domain-subset training sets $\times$ learning rate $\{3{\times}10^{-4}, 10^{-3}\}$ $\times$ seeds $\{0, 1, 2\}$. The six training sets share one

source-task-disjoint split, the same candidates, the same full five-domain validation set (60 groups per domain), and the same held-out set (120 groups per domain). They differ only in the training-domain subset: `all_core` uses coding+reasoning+math+logic+alignment (3,713 pair differences); `all_core_balanced` uses the same five domains with a seeded subsample equalized to 250 pairs per domain (1,250 total); `code_reasoning` uses coding+reasoning (1,714); `math_logic` uses math+logic (1,749); `alignment_only` uses alignment (250); and `code_math_logic` uses coding+math+logic (2,662). Pairwise training examples are reward-ordered within-group candidate differences, capped at 24 per group. Only `all_core_balanced` depends on the seed through its subsampling; with zero initialization and full-batch optimization the other five sets are deterministic across seeds, so the 36 executed runs comprise 16 distinct optimization trajectories. Each run uses Adam with weight decay 0.01, gradient clipping at 1.0, and 60 epochs, with no learning-rate scheduler or early stopping; the winning run is selected on validation macro top-1 only. A single-cell tap receives a 2048-dimensional feature vector; the cached per-candidate feature tensor across all captured loci is $[5, 4, 2048]$.

B FULL CROSS-LOOP MATRICES

Data. Bounded deterministic subset of the corrected CoreContent v2 dataset: 250 train / 60 validation / 120 held-out groups per core domain (1,250 / 300 / 600 groups; 8,592 candidates), on the existing task/prompt-disjoint split with **zero** task-ID crossings asserted; 195 `hendrycks_math` task IDs spanning splits excluded. **Features.** One frozen forward per candidate (`use_cache=False`, four UT steps, no early exit) at layers $\{8, 16, 24, 36, 47\}$ $\times$ loops {L1–L4}, mask-valid mean pooled, stored $[5, 4, 2048]$ fp16. **Tap and selection.** `AntisymLinearNoNorm`; 36 runs from the six domain-subset training sets defined in Appendix A $\times$ learning rate $\{3{\times}10^{-4}, 10^{-3}\}$ $\times$ seeds $\{0, 1, 2\}$, using Adam, weight decay 0.01, gradient clipping 1.0, zero initialization, 60 epochs, full-batch optimization, no scheduler and no early stopping; only the balanced training set changes across seeds, yielding 16 distinct optimization trajectories; selected on validation macro top-1 only. **Uncertainty.** Task-clustered bootstrap, 10,000 draws, seed 20260720, identical draws for both members of every paired comparison; matched chance (tie-aware, analytic) macro 0.2818 [0.2784, 0.2855]. **Timing evidence.** A contemporaneous run plan (17:57:33 on 2026-07-20) and the preserved extractor/statistics code fix mask-valid mean pooling, `EARLY_READABLE`, and the 0.03 margin before first feature extraction (18:06) and before train/eval output (23:49). The plan was not git-sealed or externally preregistered, so these choices are pre-specified rather than preregistered.

The selected configurations for the twelve transfer-bearing cells are shown in Table 4. All use seed 0; for these unbalanced training sets the three seeds are deterministic ties and strict greater-than selection therefore takes the first seed in grid order rather than identifying a seed advantage. Eleven

Table 4: Selected local-refit configurations for the twelve cells used in cross-loop transfer, plus the two late reference cells. `cr` = `code_reasoning`; `all` = `all_core`. All entries use seed 0.

Layer	L1	L2	L3	L4
8	cr / 3×10^{-4}	cr / 3×10^{-4}	all / 3×10^{-4}	cr / 3×10^{-4}
16	cr / 10^{-3}	cr / 10^{-3}	cr / 10^{-3}	cr / 10^{-3}
24	cr / 3×10^{-4}	cr / 10^{-3}	cr / 3×10^{-4}	cr / 3×10^{-4}
36 (ref)	—	—	—	all / 3×10^{-4}
47 (ref)	—	—	—	all / 10^{-3}

of the twelve transfer-bearing cells select `code_reasoning`; the only exception is 8_L3, which selects `all_core` on a validation tie. All three headline L1→L4 comparisons are therefore exact-configuration matched. Across the nine source→L4 transfers, eight share the training-domain subset and seven share the complete selected configuration. The sole training-set mismatch is 8_L3→8_L4; at 8_L3, `all_core` and `code_reasoning` tie at validation macro 0.6100 and the former is selected by grid order. The 250-pair `alignment_only` set never wins a reported cell. Across the full transfer matrix, each physical layer has an exact-configuration-matched L1-sourced loss and an exact-configuration-matched later-source transfer whose interval includes zero, so the headline first-pass-versus-later asymmetry does not depend on changing the selected training-domain subset.

The paired differences against the final-loop layer-24 refit — the contrast that defines the EARLY_READABLE label — are tabulated in Table 5; the four bold cells qualify (above chance and within 0.03 of the 24_L4 refit).

Table 5: Paired difference of each cell against the 24_L4 local refit, with task-clustered 95% CI.

Cell	Δ vs 24_L4 refit	95% CI	Cell	Δ vs 24_L4 refit	95% CI
8_L1	−0.2083	[−0.2567, −0.1600]	16_L4	**+0.0100**	[−0.0150, +0.0350]
8_L2	−0.0500	[−0.0900, −0.0100]	24_L1	−0.1900	[−0.2350, −0.1466]
8_L3	**+0.0150**	[−0.0217, +0.0500]	24_L2	−0.0500	[−0.0867, −0.0133]
8_L4	**+0.0033**	[−0.0267, +0.0350]	24_L3	−0.0083	[−0.0383, +0.0217]
16_L1	−0.1783	[−0.2250, −0.1317]	36_L4	+0.0283	[−0.0017, +0.0583]
16_L2	−0.0350	[−0.0750, +0.0050]	47_L4	+0.0017	[−0.0417, +0.0433]
16_L3	−**0.0017**	[−0.0333, +0.0283]			

The three loop contrasts all exclude zero — 8_L4 − 8_L1 = +0.2117 [+0.1633, +0.2600], 16_L4 − 16_L1 = +0.1883 [+0.1450, +0.2350], 24_L4 − 24_L1 = +0.1900 [+0.1466, +0.2350] — and the best early cell is statistically tied with the strongest reference: 8_L3 − 36_L4 = −0.0133 [−0.0467, +0.0200].

The nine intermediate transfers complete the matrix (Δ is the paired difference against the target-local refit; ρ is Spearman score correlation on identical held-out candidates). These transfer statistics, not cosine similarity between independently refit tap weights, are the basis for the paper's operational "rotation"/"stability" labels:

Table 6: Intermediate frozen transfers (the nine →L4 transfers are in Table 2).

Transfer	Target top-1	Δ vs target refit	ρ	Label
8: L1→L2	0.4667	−0.1016	0.690	ROTATED
8: L1→L3	0.4000	−0.2333	0.548	ROTATED
8: L2→L3	0.6100	−0.0233	0.895	**STABLE**
16: L1→L2	0.4717	−0.1116	0.845	ROTATED
16: L1→L3	0.4817	−0.1350	0.771	ROTATED
16: L2→L3	0.6117	−0.0050	0.954	**STABLE**
24: L1→L2	0.4833	−0.0850	0.845	ROTATED
24: L1→L3	0.5233	−0.0867	0.771	ROTATED
24: L2→L3	0.6100	0.0000	0.918	**STABLE**

Every ROTATED transfer's Δ interval excludes zero; no STABLE transfer's does. Pairwise sign agreement runs 0.713–0.837 for ROTATED and 0.915–0.967 for STABLE transfers.

C CONTROLS IN FULL

Label-shuffle at four representative cells (8_L1 0.3500, 8_L4 0.4267, 16_L2 0.4033, 24_L4 0.3367); text-length ranking baseline macro 0.3233 (best-signed 0.3700); extraction-path validation against the production cache on 2,434 identical held-out candidates (per-cell **mean** cosine 0.9982–0.9991 at layers 24/36/47 — the per-candidate minimum is much lower and is not claimed as a bound — with the locked policy scoring 0.6867 cached versus 0.6850 fresh); 20/20 targeted tests pass, covering scoring equivalence against the locked scorer, candidate/label alignment, split integrity, frozen-head immutability, deterministic refit, validation-only grid selection, and bootstrap clustering.

D A CROSS-ROLE BOUNDARY

The same taps were applied, as an explicitly exploratory secondary readout, to the S3B2 generated-branch-correctness pool of Kirin (2026b) (16 tasks, 160 saved branches, leave-one-task-out; coding contributes zero positives, so this is under-powered by construction). Two things follow, and they point in different directions. Locally refit at each cell, the loop trend partly reproduces: at physical

layer 16, pooled AUROC rises from 0.597 at loop 1 to 0.654 and 0.651 at loops 3 and 4. It does not reproduce at layer 24, where loop 1 (0.635) exceeds loop 2 (0.560), and layer 8 stays weak throughout (0.490–0.577), so we describe the corroboration as layer-specific rather than general. More informatively, the **frozen** CoreContent direction does not transfer to generated branches at all: transplanted AUROC lies between 0.354 and 0.475 at every one of the fourteen cells, at or below chance. This is consistent with the characterization in Kirin (2026b) of the CoreContent tap as a corruption/quality detector rather than a plausible-wrong-branch ranker, and we read it as a role and distribution boundary — the direction that ranks constructed candidate families is not the direction that separates correct from incorrect *generated* branches — rather than as a contradiction of the primary result, which concerns a different target on a different pool.

E EVALUATION-INTEGRITY PROTOCOL

The unit of independence is the *source task*, not a constructed candidate row. Source-task IDs are assigned to train/validation/held-out partitions before candidate rows are formed; the pipeline asserts that the intersection of task IDs across partitions is empty and refuses the analysis otherwise. The localization subset uses the existing corrected task/prompt-disjoint split, with 195 `hendrycks_math` task IDs that crossed the historical boundary excluded. Uncertainty is clustered at the same source-task level.

Pairwise readers must not obtain accuracy from presentation order. The primary tap here is structurally antisymmetric, $s(A, B) = w^\top (h_A - h_B)$ with no bias, so swapping the pair negates the score exactly up to numerical tolerance. For any non-structurally-antisymmetric pairwise scorer used in the broader audit (Kirin, 2026b), both presentation orders are scored and the relational component $[s(A, B) - s(B, A)]/2$ is evaluated rather than a canonical fixed-order score. These split and order checks address distinct failure modes; passing one does not substitute for the other.

F REPLICATION RUNS: FULL TABLES

Written plans survive for both replication campaigns (`family_xloop_v3_20260726/EXPERIMENT_PLAN.md`, `huginn_probe_v3_20260726/EXPERIMENT_PLAN.md`). The Huginn plan predates its feature extraction; the family plan was written after base-2.6B extraction had already begun, so only the former is described here as preregistered. Family: the identical hash-verified 2,150-group subset, tap class, 36-run selection sweep, validation-only selection, and bootstrap seed 20260720 re-run on ByteDance/Ouro-2.6B (base, pinned 1ed0425), Ouro-2.6B-Thinking (pinned f1edd81), and Ouro-1.4B (24 layers; capture layers mapped proportionally to $\{4, 8, 12, 18, 23\}$; its own tokenizer and fresh taps). The 2.6B 36_L4/47_L4 entries are independent same-protocol local refits, not copied RLTT references. Every L4−L1 loop contrast excludes zero on every checkpoint; frozen RLTT taps score within 0.028 macro top-1 of each 2.6B sibling's local refits with median Pearson/Spearman score correlations of 0.975/0.973 on Base and 0.996/0.995 on Thinking. A selected-configuration audit shows more heterogeneity than in the headline RLTT run: L1→L4 uses the same training-domain subset at 3/3 layers in Base and 2/3 in Thinking and 1.4B, while only 1/3 L1→L4 comparisons are exact-configuration matched in each replication. This does not affect the local-refit readability trend; the transfer replication is therefore treated as qualitative support. Base nevertheless has, at every tested layer, an exact-configuration-matched L1-sourced transfer to some later loop with a significant loss; 1.4B does as well, and Thinking has such exact-configuration examples at layers 8 and 24 with a training-set-matched LR-only difference at layer 16. `alignment_only` never wins a reported replication cell. Huginn: tomg-group-umd/huginn-0125 (bf16), one frozen forward per candidate at eight recurrence steps, states captured at the last recurrent-core block, masked-mean pooled $[1, 8, 5280]$; the same trainer runs on the step axis unchanged rather than using literal Ouro L4_36/L4_47 rows. Step-wise refits 0.332/0.320/0.303/0.362/0.383/0.378/0.402/0.438; step-8−step-1 +0.107 [+0.058, +0.157]; step-1-sourced taps transfer at parity through step 4; significant losses appear at L1→5 (−0.060), L1→8 (−0.093 [0.045, 0.142] in loss magnitude), and L2→8 (−0.090), while step-4-sourced taps remain within 0.027 of later local refits. The selected training subset changes in 11/17 Huginn transfer pairs, so these transfer dynamics are reported descriptively rather than used for the cross-architecture coordinate claim. Each run directory ships `SHA256SUMS` and `ENVIRONMENT.json`.

Table 7: Complete local-refit matrices for the 2.6B checkpoints (held-out macro top-1; chance 0.2818; identical subset and procedure).

Cell	RLTT	base 2.6B	Thinking 2.6B	Cell	RLTT	base 2.6B	Thinking 2.6B
8_L1	0.4100	0.4150	0.4083	24_L1	0.4283	0.4200	0.4233
8_L2	0.5683	0.5533	0.5667	24_L2	0.5683	0.5667	0.5633
8_L3	0.6333	0.6083	0.6233	24_L3	0.6100	0.6100	0.6133
8_L4	0.6217	0.6300	0.6133	24_L4	0.6183	0.6367	0.6267
16_L1	0.4400	0.4300	0.4217	36_L4	0.6467	0.6400	0.6317
16_L2	0.5833	0.5683	0.5783	47_L4	0.6200	0.5883	0.6083
16_L3	0.6167	0.6333	0.6183				
16_L4	0.6283	0.6483	0.6233				

Table 8: Ouro-1.4B (24 layers; mapped cells; its own tokenizer and fresh taps).

Layer	L1	L2	L3	L4
4	0.4650	0.5250	0.6017	0.6033
8	0.4617	0.5550	0.6100	0.6083
12	0.4733	0.5783	0.6233	0.6100
18 (ref, L4 only)	—	—	—	0.5683
23 (ref, L4 only)	—	—	—	0.5600

Table 9: Huginn-0125, recurrent-core boundary (chance 0.2818): step-wise local refits and step-1-sourced frozen transfers (Δ against the target-step refit). In the L1-sourced row shown, * marks intervals excluding zero: L1$\to$5 [-0.110, -0.012] and L1$\to$8 [-0.142, -0.045]; the full transfer ledger additionally marks L2$\to$8 ($\Delta = -0.090$) as excluding zero. Step-4-sourced taps transfer forward at $\leq$0.027 loss throughout.

Step	1	2	3	4	5	6	7	8
Local refit	0.3317	0.3200	0.3033	0.3617	0.3833	0.3783	0.4017	0.4383
L1$\to$step Δ	—	+0.017	+0.017	-0.018	-0.060*	-0.023	-0.037	-0.093*