TECHNICAL REPORT · CCAI AGENTS SEMINAR

REVIEW AND TESTS OF ARXIV:2611.05234

ARLoop: attention over depth for looped transformers

review and tests in the recurrent-proof regime

Bartosz Naskręcki

Faculty of Mathematics and Computer Science, Adam Mickiewicz University, Poznań

Centre for Credible AI, Warsaw University of Technology

six parts re-checked

Stage 4: re-checked

Stage 2: re-checked

BUILT 2026-11-17 01:31 CET FROM THE REPOSITORY AT COMMIT 12bbf1d

Contents

Summary 4

1 The question 6

2 The paper 7

2.1 Looped transformers and their residual stream 7

2.2 The released models, as configured [code] 8

2.3 The claims, and what we did with each 8

3 Methods 11

3.1 Paired comparisons 11

3.2 Protocols before evaluations 11

3.3 The independent re-check 12

3.4 Evidence tiers and the research map 12

3.5 Compute, receipts and numbers 13

4 E1 · the released checkpoints 13

4.1 Design 13

4.2 The headline gap, and its dependence on BOS (E1.1, E1.2) 14

4.3 More loops than trained (E1.3) 15

4.4 One loop past the trained depth (exploratory) 16

4.5 Where the routing looks (E1.4, descriptive) 17

4.6 Re-check 18

5 E2 · the canary 18

5.1 Design 18

5.2 Results 20

5.3 Re-check 20

6 E3 · checked proofs 21

6.1 Design 21

6.2 Results 22

6.3 Re-check and limits 22

7 Stage 1 · training more 22

7.1 Proofs: E3's pair trained eight times longer (1a) 23

7.2 The canary trained 10× and 30× longer (1b) 24

7.3 A trained fourth loop, routing only (1c) 25

7.4 What Stage 1 says 27

8 Stage 4 · confirmatory proofs 27

8.1 Design 27

8.2 Results 30
9 Stage 2 · training at depth 34
9.1 Design 34
9.2 Results 36
10 Findings register 40
11 Discussion 42
12 Limitations 44
13 Reproducibility 45
A Every paired comparison 47
A.1 E1: every paired comparison 47
A.2 E2 (1,000 updates): every comparison 47
A.3 Stage 1b (10,000 updates): every comparison 50
A.4 Stage 1b (30,000 updates): every comparison 53
A.5 E3: exploratory paired diagnostics 56
A.6 Stage 1a: the full ladder 57
A.7 Stage 1c: rules 1 and 2 57
B The protocols in brief 57
C The research map's test cells 59
D Glossary 60
E References 61
Colophon 62

SUMMARY

What we found, *in one page*

This report is the complete account of a seminar study of ARLoop (Yao, Du, Zhu and Sun, arXiv:2611.05234). We read the paper and its released code, re-measured three of its seven claims, fully or in part, on the released checkpoints, and tested its idea in the regime of our programme `latent`, where looped models write proofs that a checker accepts or rejects.

ARLoop replaces the fixed sum of a looped transformer's residual stream by attention over depth, and the paper argues that this residual design affects how much later passes are used. Our question: what, in practice, decides whether an extra loop helps?

+5.16

GSM8K points, ARLoop-0.6B over PreNormLoop-0.6B with BOS, as the authors evaluate; 95% interval [+2.65, +7.66]

0 of 8

paired comparisons beyond the trained 3 loops in which a released model gains

4 and 3/32

dev proofs accepted at 3 loops after 4,096 LoRA updates (ARLoop, PreNorm); 0 and 0 after 512

- **The released checkpoints (E1).** The pair's GSM8K gap reproduces with BOS: 38.97% against 33.81%, +5.16 points [+2.65, +7.66]. Without BOS it is inconclusive, +2.20 [−0.45, +4.85]: the gap's size depends on one evaluation detail (finding F9). ARLoop-1.2B scores 63.00% (paper 63.1).
- **More loops than trained (E1).** No gain is detected beyond the trained 3 loops: a released model gains significantly in none of the 8 paired comparisons. Below 3 loops all 5 runs score 0%. One loop past the trained depth ARLoop-0.6B loses less than PreNormLoop-0.6B (exploratory): 10.39 points under `repeat` and 0.15 under `cycle`, against 17.66; under `cycle` a descriptive probe is consistent with its read-out routing around the extra loop.
- **The canary (E2).** Tiny looped models trained from scratch on a Lean-checked task: no candidate effect in any of the 6 primary comparisons; no out-of-distribution depth item solved by any arm.
- **Checked proofs (E3).** After 512 LoRA updates the matched pair is at the proof floor: 0 of 32 dev targets in all 16 suites, while the gold proofs pass the same harness.
- **Training more (Stage 1a).** With 4,096 updates both bases leave the floor (4 and 3 of 32 at 3 loops). Trained at random depths, ARLoop-0.6B proves 0, 2, 3, 3 targets at 1–4 loops and PreNormLoop-0.6B none. The registered test, on valid single steps (61 against 55 of 64 rows at R = 4 and R = 1), misses after Holm correction ($p = 0.0625$): a one-seed lead for ARLoop, below the registered threshold.
- **Training longer or routing only (Stage 1b, 1c).** The canary trained 10× and 30× longer gains accuracy, and no depth effect is detected. A trained fourth loop in ARLoop-0.6B gives no GSM8K gain (−0.61 and −0.38 points against matched controls); the frozen read-out keeps 92–94% of its weight on loop 3.
- **Findings about the paper.** 11 findings (F1–F11): wording, scope, an inert parameter group, and a headline gap that depends on BOS. None changes a number in the paper's tables.
- **Confirming the lead (Stage 4).** With three seeds, on the sealed test split of latent's Campaign 4 opened once: ARLoop's random-depth adapters make 64 single steps valid only at R = 4 against 1 only at R = 1, of 672 paired rows (Holm $p = 7.2 \times 10^{-18}$; 3 of 3 seeds favour R = 4): more loops help ARLoop, as registered. At the trained depth the pair shows no detectable difference in test proofs (4 against 5 discordant goals, Holm $p = 1$). ARLoop's gain from one to four loops minus PreNorm's (secondary): +4.32 points [+1.19, +7.44], ARLoop gains more. PreNorm's random-depth adapters gain from more loops too (S1, secondary: 51 against 17 single steps, McNemar $p = 4.5 \times 10^{-5}$), less than ARLoop's: the effect is not ARLoop's alone.
- **Training at depth (Stage 2).** Every parameter of both 0.6B models trained further on 0.1B tokens of math text (3,051 updates, one seed), at R = 3 and at R = 4: no gain is detected for ARLoop-0.6B and PreNormLoop-0.6B (ARLoop-0.6B +0.91 points [−0.61, +2.43], Holm $p = 0.55$; PreNormLoop-0.6B +0.61 points [−1.29, +2.58], Holm $p = 0.59$). ARLoop's difference minus PreNorm's (secondary): +0.30 points [−2.20, +2.73], not established. The training itself costs accuracy at R = 3, against the released models: ARLoop-0.6B −4.62 points [−6.82, −2.43]; PreNormLoop-0.6B −3.64 points [−5.69, −1.52]. PreNorm was also trained with a random number of loops per update, 2 to 5 (rule 6, descriptive; ARLoop's random-depth arm did not run): GSM8K accuracy 6.97%, 24.26%, 25.32%, 24.03% at R = 2, 3, 4, 5; R = 5 against R = 3, −0.23 points [−2.35, +1.90].

THE ANSWER SO FAR

So far, **the trained depth bounds what the models do**: every released model collapses below its trained depth and no gain is detected above it, and more training lifts the proof models off the floor. **One loop past the trained depth, ARLoop loses less than the PreNorm loop** (exploratory); under the `cycle` rule a descriptive probe, not re-checked, is consistent with its read-out routing around the extra loop, while under `repeat` it still loses 10.39 points. ARLoop can turn extra loops into accuracy when it is trained for them: a one-seed lead from random-depth training on proofs, which Stage 4 confirmed with three seeds on unseen goals; the PreNorm loop gains too, less (secondary). With every parameter trained at four loops (Stage 2, one seed), no gain is detected for ARLoop-0.6B and PreNormLoop-0.6B.

What decides whether *an extra loop is used?*

Our `latent` programme found that extra passes of a retrofitted recurrent Qwen rarely helped checked proofs, and that learned stopping collapsed to one pass. ARLoop claims that the *residual design* affects how much later passes are used. That is testable on our tasks.

The released matched pair (ARLoop-0.6B and PreNormLoop-0.6B: same data, tokeniser, backbone and budget; only the residual path differs) makes a controlled test possible.

When we ask whether looping *pays off*, we mean one of two things, fixed before the Stage 1 runs (`CAMPAIGN.md`), each measured paired on the same items:

1. **Within a model:** in a model trained to work at several depths, accuracy rises with the number of loops.
2. **Between models:** a model trained at more loops beats one trained at fewer, on the same data and the same number of updates or tokens.

TABLE 1 The study at a glance. E1–E3 were planned with the talk; Stage 1 asked whether more training changes their answers; Stages 4 and 2 are the follow-ups.

PART	QUESTION	MODELS	MEASURE	STATUS
E1	Do the paper's numbers hold on the released weights; do the models gain from more loops; where does the routing look?	the three released checkpoints	GSM8K, WikiText-2	RE-CHECKED
E2	Does attention over depth change how many Lean-checked certificates a tiny looped model writes?	tiny decoders trained from scratch, six arms, three seeds	Lean-checked certificates	RE-CHECKED
E3	Does ARLoop's residual help the matched 0.6B pair write checked proofs after identical fine-tuning?	ARLoop-0.6B and PreNormLoop-0.6B with LoRA	checked proofs, single steps	RE-CHECKED
1a	Does eight times the training lift E3's pair off the floor, and do more loops then help?	as E3	checked proofs, single steps	RE-CHECKED
1b	Does the canary use depth when trained 10× and 30× longer?	as E2	Lean-checked certificates	RE-CHECKED
1c	Does a fourth loop pay off once its routing is trained?	ARLoop-0.6B, routing only	GSM8K, WikiText-2	RE-CHECKED
Stage 4	Does the 1a lead hold with more seeds on the sealed test split?	as 1a	checked proofs, single steps	RE-CHECKED
Stage 2	Does training the whole model at four loops make a fourth loop useful?	the matched pair, all parameters	GSM8K, WikiText-2	RE-CHECKED

How to read this report

- Claims of the paper are tagged as in the digest: **[paper]** taken from the paper, **[code]** read in the released code or checkpoints, **[exp]** re-measured by us, **[num]** a small numerical check.
- Colours keep one meaning throughout: `ARLoop` attention over depth, `PreNorm` the fixed sum of updates, `checked` accepted by a checker or re-checked, `finding` a finding about the paper or a failure. Everything else is neutral.

- We write R for the number of loops, also for E2 (whose code calls it r) and for E3 and Stage 1a (whose code calls it L); quotations of their protocols keep their letters.
- Abbreviated references (§, Tab., Fig., App., eq.) point into the paper; a § after a file name points into that file. This report's own parts are written out: section, Table, Figure, Appendix. Quotations from the digest say so in their label.
- Every number is read from a result file, a protocol or the digest when this report is built; a sentence whose files no longer support it stops the build.

2 THE PAPER

Attention over depth, *and seven claims*

Yao, Du, Zhu and Sun, *ARLoop: Fully-Open Pre-trained Looped Transformers with Attention Residuals*, arXiv:2611.05234v1, submitted 2026-11-03, 26 pages. The full digest, with every claim tagged and the findings register, is published with the notes.

2.1 Looped transformers and their residual stream

A looped transformer reuses one stack of layers several times. Its parameters are those of the stack; its depth is the stack times the number of loops R. In a loop the residual stream is the only memory that carries information from one pass to the next, and in the usual PreNorm design every update is simply added to it:

$$h_i = h_{i-1} + f_i\big(\mathrm{Norm}(h_{i-1})\big) \quad\Longrightarrow\quad h_i = h_0 + \sum_{j\le i} z_j, \qquad z_j = f_j\big(\mathrm{Norm}(h_{j-1})\big)$$

The PreNorm residual of sublayer i (attention or MLP), unrolled: every update z_j enters every later state with the same weight. A looped transformer applies the same stack of sublayers R times, so the stream accumulates R rounds of updates from the same weights.

$$h_i = \sum_j \alpha_{i,j}\, s_j, \qquad \alpha_{i,\cdot} = \mathrm{softmax}_j\left(q_i^{\top}\,\mathrm{Norm}(s_j)\right)$$

Attention over depth (the paper's eq. 2): the state h_i after sublayer i, which the next sublayer reads, is a softmax mixture of the sources s_j (the embedding and the residual blocks), weighted by a learned routing query q_i of sublayer i.

Looped models go back to the Universal Transformer [8]; recent ones include Huginn [3], Ouro [4], Parcae [5], DeepLoop [11] and a recurrent model retrofitted from OLMo-2 [6].

ARLoop takes Attention Residuals (Kimi Team) [2] and adapts them to loops. Updates inside a *block* of consecutive sublayers add up as in PreNorm; across blocks, attention decides. Only a *window* of sources is kept: the embedding and the 3 most recent blocks. The routing query is learned **per effective sublayer**, so it is not shared across loops even though the transformer blocks are.

2.2 The released models, as configured [code]

ARLoop-0.6B has 18 physical layers looped 3 times: 54 effective layers, 108 sublayers. A block is 18 sublayers (9 layers), committed at effective layers 9, 18, 27, 36, 45: two blocks per loop, and the sixth block is never committed (it stays the partial block that the final query reads). The routing parameters are 218 vectors of width 1,536 (a query and a norm weight per sublayer, and the read-out's): 334,848 parameters, **exactly** the difference between ARLoop-0.6B (583,138,944) and PreNormLoop-0.6B (582,804,096). ARLoop-1.2B has 24 layers × 3 loops = 72 effective layers (144 sublayers) and 593,920 routing parameters (290 × 2,048). The last routing call, before the output head, is the *read-out*.

2.3 The claims, and what we did with each

C1 · [PAPER] · PAPER §4.1, TAB. 1

ARLoop has the lowest perplexity among residual designs

Validation perplexity (10M FineWeb-Edu tokens; by App. A.1, 4,096 sequences of 2,048 tokens = 8.4M [num]), fixed-R training; learning rate swept for PreNorm and Sandwich-Norm only, ARLoop and Parcae reuse PreNorm's rate and DeepLoop reuses Sandwich-Norm's (App. A.1).

Gain over PreNormLoop at 140M: −0.11 (R = 1), −0.53 (R = 3), −0.50 (R = 5); at 370M: −0.23 (R = 1), −0.31 (R = 3) [num, from the table]. The paper's "larger gains at deeper recurrences" holds against R = 1 but is not monotone (finding F7). One number per cell: the paper states no seed count and no repeat runs, and reports no intervals. Not re-run by us (≈12k GPU-h for the controlled study, App. F).

TABLE 2 The paper's Tab. 1: validation perplexity (lower is better), transcribed in the digest; bold marks the lowest value of each column.

MODEL	140M R = 1	R = 3	R = 5	370M R = 1	R = 3
Sandwich-Norm	18.98	17.19	16.59	13.92	12.97
DeepLoop	18.72	17.05	16.52	13.69	12.78
Parcae w/o injection	18.37	16.60	16.01	13.55	12.31
Parcae with injection	18.37	16.55	15.96	13.55	12.30
PreNormLoop	18.37	16.91	16.34	13.55	12.48
ARLoop	**18.26**	**16.38**	**15.84**	**13.32**	**12.17**

C2 · [PAPER] · PAPER §4.1, TAB. 2

A sliding window is enough for loops

Read row by row [num]: the window has the lower perplexity in 5 of 6 rows but the higher downstream average in only 3 of 6; full history wins both R = 4 downstream rows. Explanation offered: depth attention is local (at sublayer 47 of the 140M R = 4 full-history model, the 3 most recent blocks + embedding get 96% of the mass) and residual blocks across loops are highly similar; in an untied stack of the same depth they are not, and the window hurts there (App. A.5, Tab. 9). Condition [paper: App. A.3, Tab. 7, and App. A.1; finding F11]: the window's edge needs value residuals. Without them (140M, R = 3) the sliding window reaches 18.20, no better than PreNorm's 18.19, while full history keeps a lead (17.52); App. A.1 says so itself: "removing value residuals substantially degrades our sliding-window depth-attention variant".

Re-measured in part on the released model (E1.4, section 4.5 of this report).

TABLE 3 The paper's Tab. 2: sliding window against full history (validation perplexity, lower is better; downstream average accuracy in %, higher is better), transcribed in the digest; bold marks the best value of each row, for each metric.

SCALE	R	PPL PRENORM	FULL HISTORY	WINDOW	DOWNSTREAM PRENORM	FULL	WINDOW
140M	2	17.48	**17.27**	17.30	42.22	41.96	**42.27**
140M	3	16.91	16.46	**16.38**	42.72	**43.69**	43.09
140M	4	16.59	16.07	**16.03**	43.27	**44.09**	43.73
370M	2	12.83	12.68	**12.55**	48.98	48.70	**49.25**
370M	3	12.48	12.27	**12.17**	49.29	50.23	**50.67**
370M	4	12.27	12.01	**11.99**	50.19	**50.93**	50.76

C3 · [PAPER] · PAPER §5, TAB. 3

The gains transfer to realistic pretraining

SmolLM-style 3-stage recipe (FineWeb-Edu, FineMath, Stack-Edu, ...; stage boundaries at 60% and 80%), GQA + SwiGLU, context 4,096. lm-eval 0.4.12 (vLLM backend), BF16, greedy ≤ 1,024 new tokens, no chat template; GSM8K = `gsm8k_cot` 5-shot flexible-extract [paper: App. B, Tab. 13; code: `lm_eval_tasks/arloop_largescale/arloop_largescale.yaml` l. 15–16, `pyproject.toml` l. 41, `README.md` l. 144 (vLLM), `scripts/evaluate/evaluate_all_benchmarks_vllm.py` l. 19–24].

(Gemma 3-4B, OLMo 2-7B, OLMo 3-7B and Huginn at 0.18T omitted here.) Single checkpoints, no intervals. Ouro-1.4B (open weights, not open data) is ahead on 7 of 8 tasks; the paper attributes this to data.

Re-measured for GSM8K (E1.1, E1.2), with a BOS control (finding F9). Ouro-1.4B's lead on 7 of 8 tasks is over ARLoop-1.2B.

TABLE 4 The paper's Tab. 3 (a selection, as transcribed in the digest): FLOPs in units of 10^{21}, training tokens in trillions, benchmark accuracies in %.

MODEL	LOOPS	FLOPS (1E21)	TOKENS (T)	ARC-C	HELLA	WINO	OBQA	GSM8K	MATH	MBPP	HUMANEVAL
Qwen3.5-2B	–	432	36	54.3	67.5	66.4	37.8	60.6	35.8	49.5	32.9
Ouro-1.4B	4	259	7.7	59.9	74.4	70.5	41.4	78.2	70.5	68.3	52.4
Huginn-3.5B	32	127	0.8	44.2	68.0	62.4	40.0	26.1	14.1	33.3	20.7
PreNormLoop-0.6B	3	2	0.18	46.3	60.6	61.2	39.8	35.7	20.9	33.1	17.6
ARLoop-0.6B	3	2	0.18	45.5	62.5	61.8	40.8	40.9	26.3	36.2	20.1
ARLoop-1.2B	3	10	0.5	56.3	71.5	66.8	45.5	63.1	37.8	52.1	33.1

C4 · [PAPER] · PAPER §5.1, FIG. 4; APP. C

Later loops contribute more in ARLoop

Layer-effect metric of Csordás et al. on 50 MATH items: layers of loops 2–3 change later representations more in ARLoop-0.6B than in PreNormLoop-0.6B.

Not re-run (the plan's optional E1.5).

C5 · [PAPER] · PAPER §6, APP. D.1

Fixed recurrence beats Poisson-sampled recurrence at R = 4

140M: PPL 16.07 (fixed) vs 16.69 (per-sequence Poisson) vs 16.52 (per-batch); 370M: 12.01 vs 12.33 vs 12.36. Per-batch runs get more tokens to match FLOPs; fixed and per-sequence runs use the same tokens. The clamped draw $\boldsymbol{R = \min(4, \max(1, \tilde{R}))}$, $\boldsymbol{\tilde{R} \sim \text{Poisson}(4)}$, gives $\boldsymbol{R = 4}$ with probability 0.567 and mean depth 3.24 [num]. Relevant to `latent`, whose random-depth arm used {1, 2, 4}.

Not re-run.

C6 · [PAPER] · PAPER §6, APP. D.2

No extrapolation beyond the trained recurrence

Setting: the per-sequence Poisson **full-history** 140M/370M models (trained R ≤ 4), evaluated at R = 1… 8; queries for unseen depths are randomly initialised and trained on 0.5B tokens with everything else frozen. PPL does not improve beyond R = 4. The released sliding-window fixed-R 0.6B/1.2B models were not swept.

Re-measured on the released models (E1.3) and, with trained routing, in Stage 1c.

C7 · [PAPER] · PAPER §6, TAB. 4

Cost

Same memory as PreNorm; training throughput −15% (full history) and −10% (window) at R = 4 on one H100 (0.6B: 28.9k → 25.9k tokens/s). The two percentages are the paper's averages over the four model sizes of Tab. 4 (§6: "approximately"); recomputed from the table [num]: −15.0% and −10.8%.

Not re-run.

Paired, registered, *re-checked*

3.1 Paired comparisons

Every comparison is **paired**: the same items, the same seeds, the same data order and, where training is involved, the same batches and the same draws of the number of loops for both residual designs. For a pass/fail outcome we report the two accuracies, their difference in points, the discordant counts (items solved by one side only) and two measures of uncertainty:

- the **exact two-sided McNemar test** on the discordant items (a binomial test with probability one half);
- a **paired bootstrap** 95% interval for the difference: 10,000 resamples of items (within each seed when there are several), random seed 20261115.

Holm's step-down correction is applied within the registered primary families of Stage 1a, Stage 1c, Stage 4 and Stage 2. E1's cells are each read with their own registered rule, without a multiplicity correction; E3's comparisons could not be read at the proof floor. E2 and Stage 1b use a discovery rule instead, registered as follows:

> REGISTERED IN `experiments/e2_small/PROTOCOL.md §6`
>
> - **Candidate effect (T4 discovery):** the 95 % interval excludes 0 **and** the per-seed differences have the same non-zero sign in all three seeds. No multiplicity correction; every comparison is reported. A candidate effect is a lead for a confirmation run (more seeds, longer training), not a finding. Anything else is "no effect detected at this scale", not evidence of absence.

3.2 Protocols before evaluations

Each experiment has a protocol committed to the repository before its first run: the question, the arms, the data, the decision rules and the compute. Changes after registration are dated amendments made before the affected results existed; analyses defined after a result was seen are labelled *exploratory* and carry no decision rule. The registered reading of each experiment is produced by an analysis script committed before its results were read.

TABLE 5 The protocols, their registration and their amendments, in CET (Appendix B lists each protocol's sections). A protocol states when it was written; the first commit records it in the repository minutes later.

PROTOCOL	REGISTERED (ITS OWN LINE)	FIRST COMMIT	AMENDMENTS
E1 · experiments/e1_release/PROTOCOL.md	2026-11-15	379b98a 2026-11-15 03:30 CET	2026-11-15 03:34 CET: BOS; 2026-11-15 03:34 CET: Sample selection; 2026-11-15 03:53 CET: BOS, corrected
E2 · experiments/e2_small/PROTOCOL.md	2026-11-15 03:31 CET	4fcdd39 2026-11-15 03:35 CET	none
E3 · experiments/e3_pair/PROTOCOL.md	with the first commit	4fcdd39 2026-11-15 03:35 CET	none
Stage 1a · experiments/e3_pair/PROTOCOL-ladder.md	2026-11-15 12:55 CET	dbef249 2026-11-15 12:56 CET	amendment 1, 2026-11-15 13:23 CET
Stage 1b · experiments/e2_small/PROTOCOL-long.md	2026-11-15 12:55 CET	dbef249 2026-11-15 12:56 CET	none
Stage 1c · experiments/e1_release/PROTOCOL-loop4.md	2026-11-15 12:59 CET	1a36937 2026-11-15 13:00 CET	amendment 1, 2026-11-15 13:56 CET
Stage 4 · experiments/e3_pair/PROTOCOL-confirm.md	2026-11-16 00:01 CET	720cab6 2026-11-16 00:06 CET	amendment 1, 2026-11-16 00:15 CET
Stage 2 · experiments/s2_depth/PROTOCOL.md	2026-11-16 00:05 CET	86c6f82 2026-11-16 00:05 CET	amendment 1, 2026-11-16 00:17 CET; amendment 2, 2026-11-16 11:41 CET

3.3 The independent re-check

A result counts as established only after a re-check with separate code that shares nothing with the analysis: GSM8K items are re-scored from the raw generations, certificates and proofs are recounted from the raw records, every Stage 1a proof attempt that took at least one action is replayed with the admitted checker, and the registered comparisons are recomputed. Each re-check writes a note (`runs/<part>/referee.md`), most with a JSON file of the recount; nothing enters the digest, the slides, the notes or this report before its note exists. Exceptions: E2's re-check recounted the certificates and took the comparisons' verdicts from the analysis file, and Stage 1b's bootstrap intervals were not recomputed (the rule's sign condition was). The re-checks were written by a different agent than the analyses; every note ends "human check pending".

3.4 Evidence tiers and the research map

Statements carry an evidence tier, from proved to speculation. All experimental results here are tier T4, computational evidence on samples; conjectures (T5) name the test that would decide them.

TABLE 6 Evidence tiers of the research map.

TIER	MEANING
T1	proved (literature or standard)
T2	proved here (not peer reviewed)
T3	exhaustive computation
T4	computational evidence
T5	conjecture
T6	hunch (speculation)

TABLE 7 The research map's planned test cells of E1–E3 and their status after the re-checks (Appendix C lists every cell and the map's definitions of the statuses; Stage 1 enters the map as evidence nodes, not as cells).

TEST FAMILY	CELLS	SUPPORTS	CONTRADICTS	INCONCLUSIVE	NOT MEASURED
E1 released checkpoints	13	12	0	1	0
E2 canary (Lean-checked)	6	0	1	5	0
E3 checked proofs (Campaign 4 dev)	8	0	0	8	0

3.5 Compute, receipts and numbers

All GPU work ran on the WMI cluster (RTX PRO 6000 nodes, and RTX 4090 for E2 and Stage 1b). Every run writes a receipt (job id, node, GPU, code commit, seeds, source hashes at start, data hashes), jobs run from read-only per-commit snapshots of the code (except E1's first jobs, which ran from a shared code directory: one of them picked up a fix mid-job, see section 4.1), and every job is recorded. The documents (digest, notes, slides, this report) are built on a laptop from those files only; each number is read or computed at build time, and a guard stops the build when a sentence no longer matches its files.

4 E1 · THE RELEASED CHECKPOINTS

The gap holds with BOS; *no model gains past its depth*

E1 re-measures the paper on the three released checkpoints, under the authors' evaluation settings with one difference (the Hugging Face backend instead of vLLM, section 4.1), and asks whether they gain from more loops than they were trained with.

4.1 Design

Models pinned at their Hugging Face revisions (ARLoop-0.6B @ `5cd28f3d`, PreNormLoop-0.6B @ `e4ad59d0`, ARLoop-1.2B @ `a29f310a`), BF16, transformers 5.23.0, lm-eval 0.4.12, one RTX PRO 6000. GSM8K [19] is evaluated with lm-eval [20] as the paper's `gsm8k_cot`: 5 shots, greedy, at most 1,024 new tokens, no chat template, flexible-extract exact match, all 1,319 test items, batches of 64. The authors' script prepends a beginning-of-sequence token (BOS) by default; our runs do the same, and each receipt records whether BOS was really prepended, measured on a probe. A first check (job 322118) found BOS ignored by the backend (128 of 128 identical generations with and without it); after the fix, job 322150's ARLoop-0.6B run, started before the fix reached the shared code directory, had no BOS and became the no-BOS control, re-run from a

frozen code snapshot as job 322258 (identical generations on 1,319 of 1,319 items); its PreNormLoop-0.6B run picked up the fix and became a repeat of the BOS run; the no-BOS PreNormLoop-0.6B run is job 322256. WikiText-2 bits per byte is the held-out language-modelling signal.

Running more loops than trained needs routing parameters (queries and norms) that ARLoop does not have. Two inference-only rules were declared before any run: `repeat` reuses the last trained loop's, `cycle` starts again from the first loop's. PreNormLoop needs no rule.

REGISTERED RULES, VERBATIM · experiments/e1_release/PROTOCOL.md

Questions and decision rules (paired over the same 1,319 items)

1. **Pair gap (E1.1, k.pairgap).** ARLoop-0.6B − PreNormLoop-0.6B at R = 3. *Supports* if the paired bootstrap 95% interval is > 0; *contradicts* if < 0; else *inconclusive*. Report exact McNemar.
2. **1.2B (E1.2).** *Supports* if within ±2 points of 63.1 (the paper's number); else *contradicts*.
3. **Loop sweep (E1.3, k.noextrap).** For each model and R > 3: *supports "no gain"* if the paired interval of (R) − (R = 3) is ≤ 0 or contains 0 with point estimate ≤ 0; *contradicts* if the interval is > 0. For R = 2 vs 3: *supports "deeper is better"* if (3) − (2) interval > 0.
4. **Routing (E1.4).** Descriptive: mean routing mass on the embedding source per effective sublayer and per loop; block cosine similarities. No decision rule (exploratory).

The protocol writes R for the number of loops, as here.

4.2 The headline gap, and its dependence on BOS (E1.1, E1.2)

TABLE 8 GSM8K, 1,319 paired items. The first row is the registered E1.1; the other three are the BOS control.

COMPARISON	A	B	A - B (POINTS)	95% INTERVAL	DISCORDANT A / B	MCNEMAR P
ARLoop-0.6B − PreNormLoop-0.6B, with BOS (E1.1)	38.97%	33.81%	+5.16	[+2.65, +7.66]	176 / 108	6.5×10^{-5}
the same, without BOS	38.44%	36.24%	+2.20	[−0.45, +4.85]	176 / 147	0.12
PreNormLoop-0.6B, with − without BOS	33.81%	36.24%	−2.43	[−4.02, −0.76]	46 / 78	0.0052
ARLoop-0.6B, with − without BOS	38.97%	38.44%	+0.53	[−1.06, +2.12]	56 / 49	0.56

With BOS the gap is reproduced: +5.16 points against the paper's +5.2; both absolute levels are about 1.9 points below the paper's numbers (produced with vLLM, according to the released README). Without BOS the gap is +2.20 points and its interval contains 0. The cause is one model: BOS lowers PreNormLoop-0.6B by 2.43 points and has no detectable effect on ARLoop-0.6B (+0.53 [−1.06, +2.12], p = 0.56). The paper evaluates every model the same way, so its table is consistent, but the size of its headline gap is not robust to this choice (finding F9). ARLoop-1.2B scores 63.00% (paper 63.1).

E1.1: SUPPORTS · E1.2: SUPPORTS · BOS CONTROL: THE GAP IS INCONCLUSIVE WITHOUT BOS

4.3 More loops than trained (E1.3)

TABLE 9 GSM8K accuracy by the number of loops; below each accuracy, the paired difference to R = 3 in points with its 95% interval. – : not run (`cycle` differs from `repeat` only beyond R = 3; ARLoop-1.2B was run where the protocol asked for it).

MODEL	R = 1	R = 2	R = 3	R = 4	R = 5	R = 6
ARLoop-0.6B, repeat	0.00% −38.97 [−41.62, −36.32]	0.00% −38.97 [−41.62, −36.32]	38.97%	28.58% −10.39 [−12.96, −7.81]	2.81% −36.16 [−38.97, −33.36]	1.74% −37.23 [−40.03, −34.42]
ARLoop-0.6B, cycle	–	–	38.97%	38.82% −0.15 [−1.97, +1.67]	–	–
PreNormLoop-0.6B	0.00% −33.81 [−36.39, −31.31]	0.00% −33.81 [−36.39, −31.31]	33.81%	16.15% −17.66 [−20.39, −14.94]	1.67% −32.15 [−34.72, −29.57]	0.61% −33.21 [−35.86, −30.63]
ARLoop-1.2B, repeat	–	0.00% −63.00 [−65.66, −60.35]	63.00%	59.21% −3.79 [−6.07, −1.52]	–	–

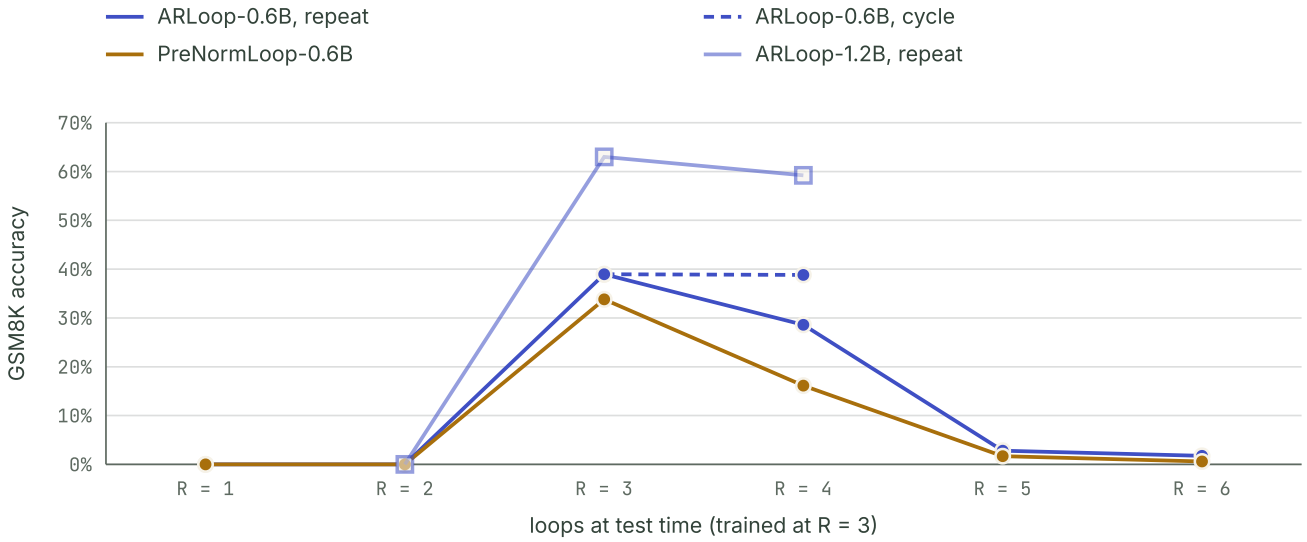

FIGURE 1 GSM8K accuracy of the released checkpoints by the number of loops (E1.3; 1,319 items each). Below the trained depth every model scores 0; above it no gain is detected. ARLoop needs a rule for the routing parameters of loops it was not trained with: `repeat` reuses the last trained loop's, `cycle` starts again from the first.

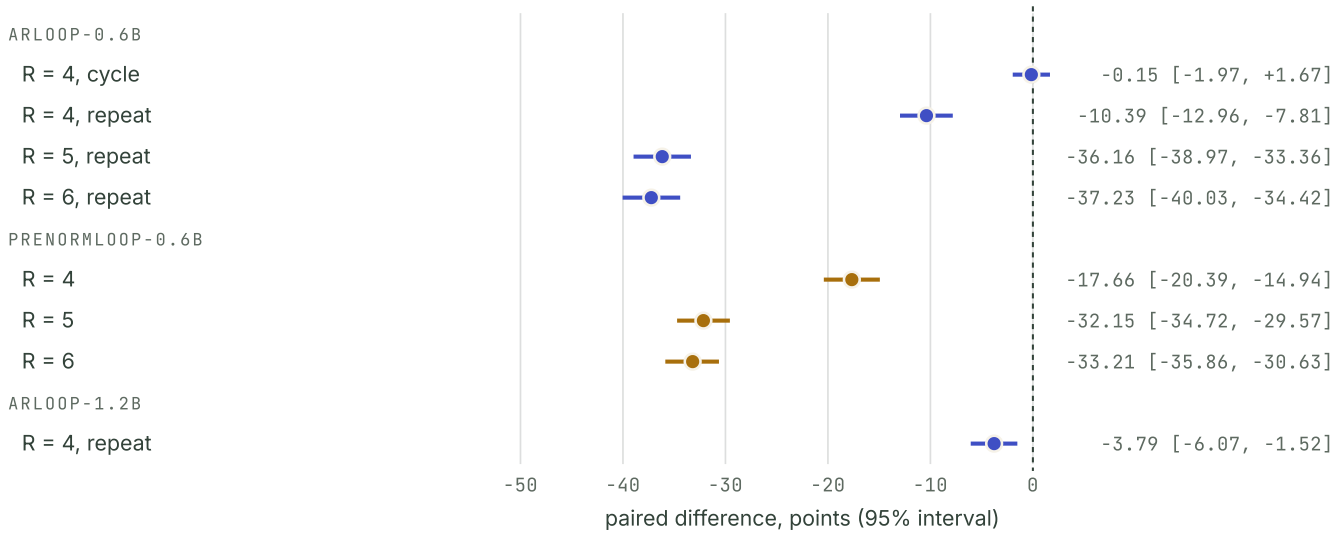

FIGURE 2 Every registered comparison beyond the trained depth (E1.3): accuracy at R loops minus accuracy at R = 3, paired over 1,319 items, with 95% bootstrap intervals. No interval lies above 0, so the decision rule reads each as "supports no gain"; an interval that reaches above 0 does not exclude a small gain.

Below the trained depth all 5 runs score 0%: the text degenerates into repetitions. Above it, no cell shows a gain; with `repeat` ARLoop-0.6B loses 10.39 points at R = 4, with `cycle` it stays level (−0.15 [−1.97, +1.67]). The released fixed-R models are not anytime models (usable at any number of loops) in either direction, and the paper's C6 holds for them too, under inference-only rules for the missing routing parameters (queries and norms). "Supports no gain" means that no gain is detected: for `cycle` at R = 4 the interval still reaches +1.67 points.

E1.3: SUPPORTS NO GAIN, IN ALL 8 CELLS (NO GAIN DETECTED)

TABLE 10 WikiText-2 bits per byte (lower is better; descriptive, no BOS). The minimum is at the trained depth for both 0.6B models. – : not run.

MODEL	R = 1	R = 2	R = 3	R = 4	R = 5	R = 6
ARLoop-0.6B, repeat	2.924	2.076	0.725	0.809	1.190	1.575
ARLoop-0.6B, cycle	–	–	0.725	0.732	–	–
PreNormLoop-0.6B	2.426	2.049	0.738	0.980	1.306	1.717
ARLoop-1.2B, repeat	–	–	0.662	–	–	–

4.4 One loop past the trained depth (exploratory)

Added after those accuracies were seen, without a decision rule: the two designs compared at the same number of loops beyond the trained depth.

TABLE 11 Exploratory: ARLoop-0.6B against PreNormLoop-0.6B at R = 4, 5, 6.

COMPARISON	A	B	A - B (POINTS)	95% INTERVAL	MCNEMAR P
ARLoop-0.6B-R4-repeat – PreNormLoop-0.6B-R4	28.58%	16.15%	+12.43	[+9.78, +15.16]	2.4×10^{-19}
ARLoop-0.6B-R4-cycle – PreNormLoop-0.6B-R4	38.82%	16.15%	+22.67	[+19.86, +25.55]	4.0×10^{-50}
ARLoop-0.6B-R5-repeat – PreNormLoop-0.6B-R5	2.81%	1.67%	+1.14	[+0.08, +2.27]	0.063
ARLoop-0.6B-R6-repeat – PreNormLoop-0.6B-R6	1.74%	0.61%	+1.14	[+0.38, +1.97]	0.0081

From R = 3 to R = 4, ARLoop-0.6B loses 10.39 points under `repeat` and 0.15 under `cycle`, PreNormLoop-0.6B 17.66. So at R = 4 ARLoop-0.6B is ahead by +12.43 points under `repeat` and +22.67 under `cycle` (Table 11), against +5.16 at R = 3: these are levels, not degradations. At R = 5 and 6 both collapse.

4.5 Where the routing looks (E1.4, descriptive)

The digest's account of the routing probe at the trained depth (64 GSM8K prompts; the quote's figure and section numbers are the paper's, and its v_j are the sources s_j of section 2.1):

> FROM THE DIGEST (ITS SECTION 4) · [EXP]
>
> - *Redundancy across loops, at matched positions.* Mean per-token cosine between residual blocks: first-half blocks b_1, b_3, b_5: 0.95 ($b_1 b_3$), 0.93 ($b_3 b_5$), 0.82 ($b_1 b_5$); second-half blocks b_2, b_4: 0.80; first vs second halves 0.0–0.1. This supports the paper's redundancy explanation (it showed the similarity for its full-history models: 140M, R = 4 in Fig. 3; 140M, R = 3 and 370M, R = 3 and 4 in App. A, Fig. 5).
> - *Norms alternate:* $\|b_1\|, \|b_3\|, \|b_5\| = 69.8, 61.3, 55.2$; $\|b_2\|, \|b_4\|, \|b_6\| = 7.7, 6.6, 6.8$; $\|h_0\| = 1.3$ (mean token norms).
> - *Embedding source:* mean softmax weight 0.143 / 0.162 / 0.131 in loops 1 / 2 / 3 (loop 1 without the inert sublayer 0, F8; 0.185 and 0.241 at the first sublayers of loops 2 and 3), but because values are mixed **before** normalisation and $\|h_0\|$ is small, its norm-weighted share $\alpha_j \|v_j\| / \sum_k \alpha_k \|v_k\|$ is only 0.034 / 0.031 / 0.023 (0.006 and 0.014 at those loop starts). Softmax weights overstate the embedding's role about fivefold; the window does not act as a strong input injection.
> - *Read-out:* 0.935 of the final softmax mass (0.897 norm-weighted) is on the last partial block.
>
> PreNormLoop-0.6B for comparison: loop-output norms 7.0, 7.4, 16.4; cosine between consecutive loop outputs 0.87 (1→2) and 0.36 (2→3).

Recomputed here from the probe's summary: the embedding's mean softmax weight is 0.143 / 0.162 / 0.131 in loops 1 / 2 / 3, and the read-out puts 93.5% of its weight on the last block. The probe at R = 4 is exploratory (added to E1's protocol after the R = 4 accuracies were seen) and was not independently re-checked. With a loop added under `cycle` the read-out still puts 93.4% on loop 3's last block and only 3.7% on the new loop's:

under `cycle` it routes around the extra loop. Under `repeat` the new loop's two blocks take 40.6% (its last block 40.0%) and loop 3's last block 57.6%. PreNormLoop-0.6B has no such switch: every update enters its stream with the same weight (Figure 3).

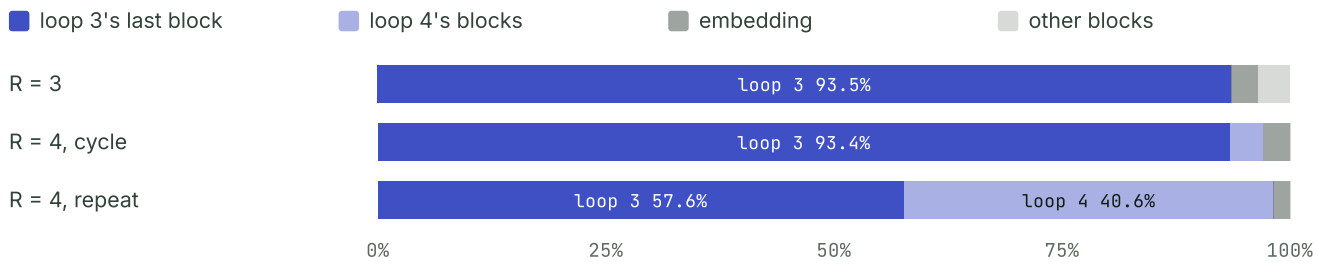

FIGURE 3 The read-out's softmax weights of ARLoop-0.6B (64 GSM8K prompts, mean over tokens): at its trained depth (descriptive), and with one loop added under each rule (exploratory, not re-checked).

4.6 Re-check

Every item of all 20 runs was re-scored from the raw generations with lm-eval's flexible-extract rule written again: 0 disagreements with the recorded scores, item ids complete in every run. Every paired comparison was recomputed with separate code: identical differences, discordant counts and McNemar p; the bootstrap intervals agree within resampling error. BOS really changes the generations (identical text on 624 of 1,319 items for ARLoop-0.6B and 526 for PreNormLoop-0.6B), while a repeated run is identical on 1,319 of 1,319. Note: `runs/e1/referee.md`.

5 E2 · THE CANARY

A tiny looped model: *no primary candidate effect*

E2 is a canary: a tiny task on which `latent` had already measured a looped decoder, repeated with ARLoop-style attention residuals in place of the loop's residual design.

5.1 Design

The task is closed addition with a certificate: the model reads `a+b=` and writes a compact certificate of the sum, which is compiled to a proof term and checked by Lean. Splits: train 93, in-distribution 35, out-of-distribution depth 64 (b in 8..11) and operand 32 (a in 16..19). Every arm is a decoder of width 64 with 4 heads, trained for 1,000 AdamW updates (batch 32, learning rate 0.001) with a random number of loops per update (1 plus a Poisson draw whose rate is lognormal with mean 3: on average 4.0 loops in training, capped at 16), on seeds 11, 29, 47; every loop arm sees the same batches and the same draws. Evaluation at $R \in \{1, 2, 4, 8, 16\}$.

TABLE 12 E2's arms (from the protocol).

ARM	THE LOOPED CORE	ROLE
latent-recurrent	latent's decoder, unchanged: a sandwich-norm block, the input re-injected by an adapter	the reference
prenorm-loop	a PreNorm block, no injection	the fixed sum
arloop-untied	PreNorm sublayers fed by attention over depth, one block per loop, a window of the embedding and the 3 most recent blocks, one query per effective sublayer	ARLoop's routing (eqs. 2–4), without value residuals
arloop-tied	the same with queries shared across loops, so any number of loops is defined	our variant
standard-parameter / -depth	unlooped controls with matched parameters or matched depth	controls

REGISTERED RULES, VERBATIM · experiments/e2_small/PROTOCOL.md

6. Comparisons and what counts as an effect

Unit of pairing: (seed, test item). For arm A against a reference arm B at the same r and split, report the accuracy difference d = mean(A − B), the per-seed count differences, the exact two-sided McNemar p on discordant units, and a 95 % paired bootstrap interval (10,000 resamples of items within each seed, RNG seed 20261115).

- **Primary family (6 comparisons):** `prenorm-loop`, `arloop-untied`, `arloop-tied` vs `latent-recurrent` on iid at r = 4 (training mean, reference recurrence) and r = 16 (deepest).
- **Secondary (descriptive):** the same at r ∈ {1, 2, 8} and on ood_depth / ood_operand; every arm vs `standard-depth`; within-arm depth change r = 16 vs r = 4 (paired over the same units).
- **Candidate effect (T4 discovery):** the 95 % interval excludes 0 **and** the per-seed differences have the same non-zero sign in all three seeds. No multiplicity correction; every comparison is reported. A candidate effect is a lead for a confirmation run (more seeds, longer training), not a finding. Anything else is "no effect detected at this scale", not evidence of absence.
- Any accepted ood_depth certificate is reported individually (latent: 0 of 128).

The protocol writes r for the number of loops.

5.2 Results

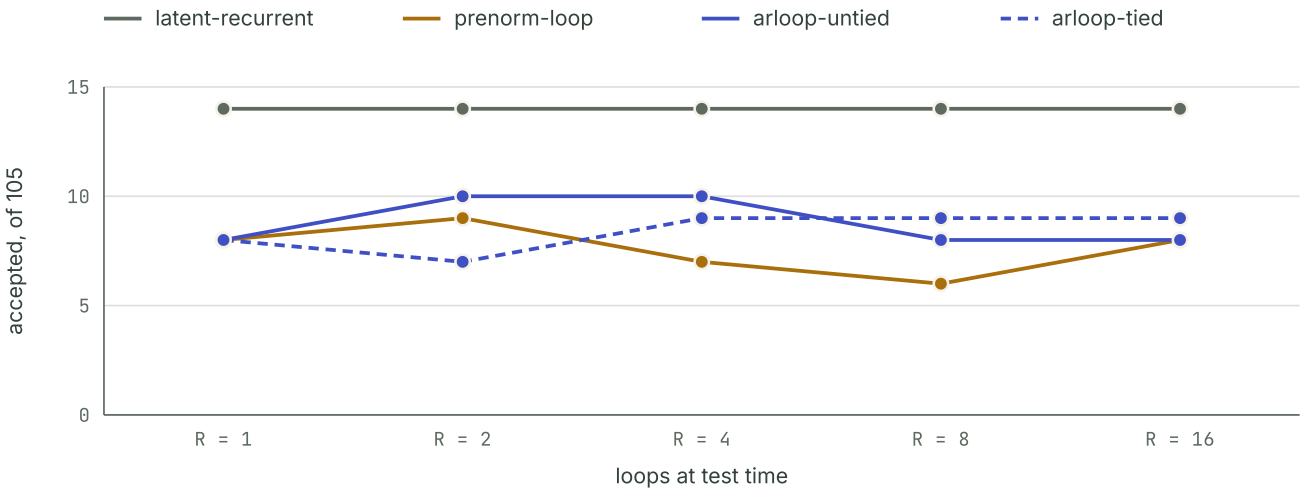

FIGURE 4 E2: Lean-accepted certificates on the in-distribution split, summed over the three seeds (of 105), by the number of loops at test time. The unlooped controls accept 11 (parameter-matched) and 10 (depth-matched).

TABLE 13 E2's primary comparisons (105 paired items: 35 per seed, three seeds).

ARM	REFERENCE	R	DIFFERENCE (POINTS)	95% INTERVAL	PER-SEED COUNT DIFFERENCES	MCNEMAR P	CANDIDATE
prenorm-loop	latent-recurrent	4	−6.67	[−13.33, +0.00]	−5 / +0 / −2	0.12	no
prenorm-loop	latent-recurrent	16	−5.71	[−13.33, +0.95]	−4 / −1 / −1	0.21	no
arloop-untied	latent-recurrent	4	−3.81	[−11.43, +3.81]	−5 / +1 / +0	0.48	no
arloop-untied	latent-recurrent	16	−5.71	[−13.33, +0.95]	−5 / +1 / −2	0.21	no
arloop-tied	latent-recurrent	4	−4.76	[−12.38, +2.86]	−5 / +0 / +0	0.36	no
arloop-tied	latent-recurrent	16	−4.76	[−12.38, +2.86]	−5 / +0 / +0	0.36	no

Every new arm is below `latent`'s decoder (−3.81 to −6.67 points) and every interval contains 0: no candidate effect in the primary comparisons (Table 13). Among the secondary comparisons there are four candidate effects: three put a new arm below `latent`'s decoder, and one puts the decoder above an unlooped control (Appendix A.2). No arm accepts a single out-of-distribution depth item at any number of loops. With tied queries, R = 16 and R = 4 give identical verdicts on every item. Within the same job, the unchanged decoder and the unlooped controls (seeds 11 and 29) reproduced `latent`'s own job 284986 record for record: 1,834 of 1,834 records identical in prediction and Lean verdict.

NO CANDIDATE EFFECT IN THE PRIMARY COMPARISONS

5.3 Re-check

The Lean-accepted certificates were recounted from the raw predictions with a separate script (every cell of the analysis table matches), and the reproduction of `latent`'s job was checked record by record. Note: `runs/e2/referee.md`.

The matched pair writing proofs: *at the floor*

E3 puts the matched pair to work on the task `latent` was built for: proofs in a constructive arithmetic, written step by step and accepted only by a checker.

6.1 Design

Both released 0.6B models are fine-tuned identically with `latent`'s Campaign 4 recipe: LoRA rank 32 (alpha 64) on every attention and MLP projection of the shared blocks, 512 updates of 16 sequences, peak learning rate 5e-5 after 32 warm-up updates, one seed (11). Two arms: **native3** trains at 3 loops; **random124** draws each update's number of loops from {1, 2, 4} (`latent`'s rule), with the same draws for both bases. ARLoop carries 36 extra routing queries (and norms) for the loop it was not trained with, copies of loop 3's, trained only in random124. Data: the proof corpus of `latent`'s Campaign 4 (C4 in the protocols quoted below, not the paper's claim C4), train 7,272 rows from 154 sessions, development 1,523 rows from 33 sessions; the test split stays sealed.

Evaluation. The 32 development proof targets at a forced number of loops R = 1, 2, 3, 4 (16 suites): one greedy trajectory per target, one action per model call, at most 96 actions, 128 new tokens per action and 12,288 decision tokens, 600 s per target. A proof is accepted when the pinned constructive Python checker replays it to a kernel-checked QED (not a Lean check). Gates before training: wrapper identity, the gold proofs replayed to QED, memory, and restart identity.

REGISTERED RULES, VERBATIM · `experiments/e3_pair/PROTOCOL.md`

Comparisons (fixed now; all descriptive at n = 32, one seed)

- P1 (primary): ARLoop vs PreNorm, `native3`, L = 3: accepted proofs, paired by target, with the exact McNemar p-value.
- P2: ARLoop vs PreNorm, `random124`, at each L in {1, 2, 4}.
- P3 (depth extrapolation): within each base, `random124` at L = 4 vs `native3` at L = 4.
- P4: accepted proofs and dev CE as functions of L for every checkpoint.

Reading rules, fixed in advance: a paired difference of at most 2 targets is "no detectable difference". "ARLoop helps" is a *signal worth a confirmatory run* only if ARLoop leads by >= 3 targets in P1 and does not trail in `random124` at L = 4. If both bases accept <= 1 target in every suite, the run is floor-limited and uninformative about the question. Dev CE differences are reported but are not proof success. No claim above T4.

The protocol writes L for the number of loops.

6.2 Results

TABLE 14 E3, each suite: accepted proofs (of 32) · mean accepted actions per attempt · valid single steps (of 64, exploratory).

CHECKPOINT	R = 1	R = 2	R = 3	R = 4
ARLoop-0.6B, native3	0 · 0.0 · 0	0 · 0.0 · 0	0 · 16.5 · 49	0 · 12.2 · 47
PreNormLoop-0.6B, native3	0 · 0.0 · 0	0 · 0.0 · 0	0 · 15.3 · 44	0 · 6.8 · 40
ARLoop-0.6B, random124	0 · 6.7 · 43	0 · 13.0 · 45	0 · 18.4 · 45	0 · 17.0 · 45
PreNormLoop-0.6B, random124	0 · 1.0 · 36	0 · 3.5 · 39	0 · 21.8 · 40	0 · 4.2 · 39

No suite accepts a proof: 0 of 32 in all 16. The harness can accept one: the gold proofs, fed through the same rollout with the same budgets, pass 32 of 32.

FLOOR-LIMITED: THE COMPARISONS ARE UNINFORMATIVE

How the attempts fail is informative. In native3 at R = 1 and 2 every attempt is malformed or truncated, the collapse below the trained depth seen in E1. Elsewhere attempts end in a rejected tactic, most often a reused name during the opening introductions. PreNormLoop-0.6B fails all 32 targets by name reuse in four cells, ARLoop-0.6B in none.

Exploratory, post hoc (the measures were defined after ARLoop's native3 suites were seen; the amendment is timestamped before any PreNorm suite ran): ARLoop's failed attempts get further in 4 cells (paired sign test, $p \le 1.2 \times 10^{-4}$ in each), and its single-step validity (one greedy action from the gold state of a development row, valid if the checker accepts the transition; on a frozen roster of 64 rows) is at least PreNormLoop-0.6B's in all six non-collapsed cells (the other two are 0 against 0). A hypothesis for a registered run (the map's conjecture `k.e3-rollouts`), not a result.

6.3 Re-check and limits

The re-check recounted all 16 suites from the raw records (the same 32 ids, task hashes, budgets and checker in each), recomputed the oracle, the exploratory diagnostics and the pairing (identical batches and loop draws for the two bases), and verified that the ported `latent` code is identical in its code to the original except for two declared changes (two other functions differ only in shortened docstrings). Limits: one seed and one training length. Declared deviations from Campaign 4: KV-cached decoding (measured on the untuned bases), 600 s per target instead of Campaign 4's 1,200 s per 32-target suite, and no checkpoint selection (Campaign 4 selected its best development-loss checkpoint; here the development split is the evaluation set). No BOS, following `latent`'s data contract. Note: `runs/e3/referee.md`.

7 STAGE 1 · TRAINING MORE

Off the floor; *a one-seed lead for ARLoop, below the registered threshold*

In E1, E2 and E3 the models were trained little or not at all for the depth being tested. Stage 1 asked whether more training changes the answer, in three parts with protocols registered before their runs, all run on WMI on 15 November 2026.

7.1 Proofs: E3's pair trained eight times longer (1a)

E3 unchanged except the schedule: the cosine runs to 4,096 updates instead of 512, and the adapters after 1,024 and 2,048 updates are kept; E3's own adapters are the first rung. Evaluation as in E3, with single-step validity on E3's frozen 64-row roster. 30.2 GPU-h.

REGISTERED RULES, VERBATIM · experiments/e3_pair/PROTOCOL-ladder.md

Decision rules (fixed now)

1. **Off the floor** (the gate for Stage 4): either base's native3 adapter after 4,096 updates accepts at least 3 of the 32 dev targets at L = 3.
2. **More loops after training** (sense 1 of `CAMPAIGN.md`), per base, on the random124 adapter after 4,096 updates:
   a. Single-step validity at L = 4 against L = 1, paired by dev row (1,523), exact McNemar test, two-sided. "More loops help" if validity is higher at L = 4 and the Holm-adjusted p (over the two bases) is below 0.05.
   b. Accepted proofs at L = 4 against L = 1, paired by target, exact McNemar. Descriptive if fewer than 5 targets are discordant.
3. **Learning curves** (descriptive): accepted proofs, accepted actions per attempt and step validity, by L, at 512 (E3), 1,024, 2,048 and 4,096 updates, per base and arm.
4. **ARLoop against PreNorm:** descriptive only, with one seed. A lead goes to Stage 4 (three seeds, the sealed test split).
5. **Exploratory analyses** are labelled as such and logged with their commit times.

The protocol writes L for the number of loops. Amendment 1 (2026-11-15 13:23 CET), before any result, states that single-step validity uses the 64-row roster, as the code always did.

RULE 1: OFF THE FLOOR

After 4,096 updates the native3 adapters prove 4 (ARLoop) and 3 (PreNorm) of 32 targets at R = 3. Along the ladder (512 → 1,024 → 2,048 → 4,096 updates): ARLoop 0, 6, 2, 4, PreNorm 0, 1, 5, 3; with one seed and 32 targets, single proofs move these counts.

TABLE 15 Stage 1a, rule 2: the random-depth adapters after 4,096 updates, 64 paired rows.

MODEL	VALID STEPS, R = 4 / 1	DISCORDANT, R = 4 / 1	MCNEMAR P	HOLM P	MORE LOOPS HELP	PROOFS, R = 4 / 1
ARLoop-0.6B	61 / 55	6 / 0	0.03125	0.0625	no	3 / 0
PreNormLoop-0.6B	51 / 49	2 / 0	0.5	0.5	no	0 / 0

RULE 2: NOT ESTABLISHED

All 6 rows on which ARLoop's R = 4 and R = 1 disagree favour R = 4, but with 64 rows the Holm-adjusted p is 0.0625, not below 0.05.

TABLE 16 Stage 1a, descriptively: the random-depth adapters by the number of loops (Appendix A.6 has every rung and arm).

RANDOM DEPTH, 4,096 UPDATES	R = 1	R = 2	R = 3	R = 4
ARLoop-0.6B: proofs, of 32	0	2	3	3
ARLoop-0.6B: valid steps, of 64	55	61	61	61
ARLoop-0.6B: dev cross-entropy, nats/token	0.281	0.242	0.235	0.233
PreNormLoop-0.6B: proofs, of 32	0	0	0	0
PreNormLoop-0.6B: valid steps, of 64	49	50	52	51
PreNormLoop-0.6B: dev cross-entropy, nats/token	0.334	0.330	0.342	0.345

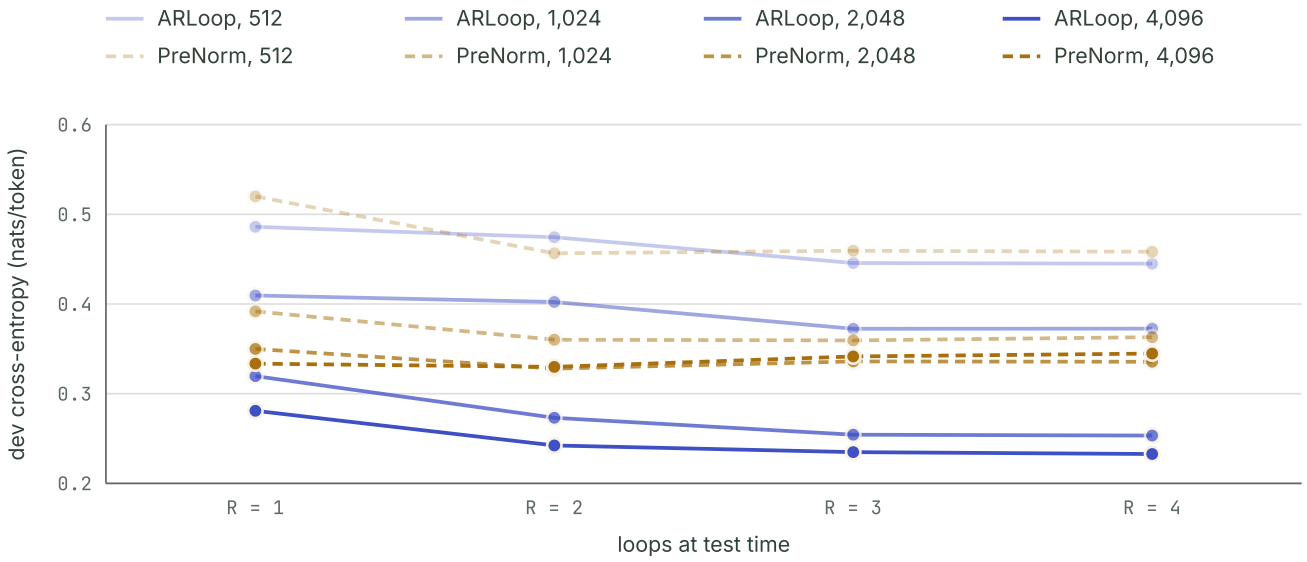

FIGURE 5 Stage 1a: development cross-entropy of the random-depth adapters by the number of loops, at 512, 1,024, 2,048, 4,096 updates (lighter to darker). ARLoop-0.6B's cross-entropy (solid, indigo) falls with more loops at every rung; PreNormLoop-0.6B's gain (dashed, gold) shrinks with training and reverses. One seed; descriptive.

Descriptively (one seed, not a registered result), ARLoop-0.6B's random-depth adapter improves on all three measures as R grows, while PreNormLoop-0.6B's proves nothing at any number of loops and its cross-entropy is higher at R = 4 than at R = 1 (Figure 5). The difference R = 1 minus R = 4 in cross-entropy, from 512 to 4,096 updates: ARLoop +0.041, +0.037, +0.066, +0.048; PreNorm +0.062, +0.029, +0.014, −0.011 nats per token.

Re-check. A separate recount of all 64 suites from the raw records: 0 disagreements. A replay with the admitted checker of every recorded attempt with at least one action (1,548 of the 2,048 attempts; the others stopped before their first action): 30 kernel-checked QEDs, exactly the 30 accepted proofs. Note: `runs/s1a/referee.md`.

7.2 The canary trained 10× and 30× longer (1b)

E2 unchanged except the number of updates: 10,000 and 30,000 instead of 1,000.

REGISTERED RULES, VERBATIM · experiments/e2_small/PROTOCOL-long.md

Decision rules (fixed now), applied to each length separately

1. **E2's primary comparisons:** each new arm against latent-recurrent, at r = 4 and r = 16 on iid. A candidate effect needs the 95% interval to exclude 0 *and* the same non-zero sign in all three seeds.
2. **Extra loops after longer training** (sense 1 of CAMPAIGN.md): within each arm, r = 16 against r = 4 on iid and on ood_depth, paired over (seed, item). Same candidate rule.
3. **ood_depth acceptance per arm and r:** descriptive. Any non-zero count is reported.
4. **Length** (descriptive): 1,000 (E2), 10,000 and 30,000 updates side by side.

The protocol writes r for the number of loops.

TABLE 17 Stage 1b: accepted in-distribution certificates at R = 4, of 105 (three seeds); the 1,000-update column is E2.

ARM	1,000 UPDATES	10,000 UPDATES	30,000 UPDATES
latent-recurrent	14	18	24
prenorm-loop	7	13	18
arloop-untied	10	16	17
arloop-tied	9	17	17

In-distribution accuracy rises from 1,000 to 30,000 updates in every arm (the arloop-tied arm stays level from 10,000 to 30,000). No depth effect is detected: none of the 16 registered within-arm comparisons of R = 16 against R = 4 (in-distribution and out-of-distribution depth, both lengths) is a candidate effect, no out-of-distribution depth item is solved, and no new arm (the PreNorm loop or either ARLoop arm) beats `latent`'s decoder (0 of 6 primary comparisons at either length). Re-check: 36 comparisons recomputed from the raw predictions, 0 disagreements (`runs/s1b/referee.md`).

The secondary comparisons do have candidate effects (10 at 10,000 updates, 7 at 30,000; Appendix A.3 and A.4): none is a depth comparison, and none puts a new arm above `latent`'s decoder.

NO DEPTH EFFECT DETECTED AT EITHER LENGTH

7.3 A trained fourth loop, routing only (1c)

ARLoop-0.6B with everything frozen except the routing of one loop. **L4** runs four loops and trains the new loop's queries and norms, which start as a copy of the loop-1 queries and norms (E1 cycle rule at R = 4); the control **C3** runs three loops and trains loop 3's the same way. Same data, order, optimiser and seed: 3,051 updates, 0.10B tokens of FineMath-4+, from which documents sharing a 13-gram with a GSM8K test question were removed (1 document). Trained parameters: 110,592 per arm.

REGISTERED RULES, VERBATIM · experiments/e1_release/PROTOCOL-loop4.md

Decision rules (fixed now)

1. **Primary:** L4 at R = 4 against C3 at R = 3, paired over the 1,319 items. Report the difference, the 95% bootstrap interval and the exact McNemar p. "A trained fourth loop pays off" if the interval lies above 0.
2. **Secondary:**
   - L4 against the released model at R = 3 (E1's `ar06-R3`);
   - L4 against the untrained `cycle` rule at R = 4 (E1's `ar06-R4-cycle`);
   - C3 against the released R = 3 (the effect of the training alone).
3. **Routing probe** (descriptive): the read-out's weight on loop 4's blocks, before and after training, with E1's `routing_compare.py` method.
4. **Exploratory analyses** are labelled as such and logged with their commit times.

Amendment 1 (2026-11-15 13:56 CET, before any result) added a second pair, L4e and C3e, with the same data, order, optimiser, schedule and seed, which trains the effective routing vector directly (55,296 parameters per arm): in the loop-4 bank 34.4% of the coordinates had query and norm weight both exactly 0, which training the two separately cannot move (finding F10). Rule 1 then applies to both pairs with Holm correction over the two; a trained fourth loop pays off if at least one pair passes after correction and the other pair's point estimate has the same sign.

TABLE 18 Stage 1c, rule 1 (1,319 paired GSM8K items).

PAIR	A	B	A - B (POINTS)	95% INTERVAL	DISCORDANT	MCNEMAR P	HOLM P	PAYS OFF
L4 – C3	37.91%	38.51%	−0.61	[−2.20, +1.06]	54 / 62	0.52	1	no
L4e – C3e	38.13%	38.51%	−0.38	[−2.05, +1.36]	62 / 67	0.72	1	no

TABLE 19 Stage 1c, rule 3: the read-out's routing weights before → after training (64 GSM8K prompts).

ARM	READ-OUT WEIGHT ON LOOP 3'S LAST BLOCK	ON THE NEW LOOP'S BLOCKS
L4	93.4% → 93.7%	3.7% → 3.3%
C3	93.5% → 93.5%	–
L4e	93.4% → 92.4%	3.7% → 4.6%
C3e	93.5% → 93.5%	–

RULE 1: THE TRAINED FOURTH LOOP DOES NOT PAY OFF

Both pairs are slightly negative and both intervals contain 0: no GSM8K loss is detected either. On WikiText (descriptive, one run) the trained loop's bits per byte fall from 0.7320 to 0.7256, close to the trained depth's 0.7247 but not below it. The read-out is consistent with the null: it is frozen by design and keeps 92–94% of its weight on loop 3's last block, before and after training (rule 3 is descriptive; its weights are read from the analysis, not recounted). This is the released-model version of the paper's App. D.2: here too, routing trained for a depth the model never saw brought no detectable gain. Re-check: every GSM8K item re-scored from the raw generations, 8 comparisons recomputed, 0 disagreements (`runs/s1c/referee.md`).

7.4 What Stage 1 says

- Training longer lifts the matched pair off the proof floor at its trained depth, in both bases.
- One sign of a result of looping, within a model and only in ARLoop: its random-depth adapter proves more, makes more valid steps and has a lower cross-entropy at more loops; PreNormLoop-0.6B's does not. With one seed and 64 rows it is a lead for Stage 4, below the registered threshold, not a result.
- No depth effect detected elsewhere: not in the canary trained much longer, and training the routing of a fourth loop does not make it pay off; the frozen read-out keeps routing around it, a likely reason that Stage 2 tests.

8 STAGE 4 · CONFIRMATORY PROOFS

Does the ladder's lead hold? *More seeds, the sealed test split*

Stage 4 turns Stage 1a's lead into a registered test: the same harness and recipe, more seeds, evaluated once on the sealed test split of latent's Campaign 4 corpus. It comes before Stage 2 because it follows directly from Stage 1a (the campaign's numbering is older than this order; the confirmatory proposal calls this run its Stage 2).

8.1 Design

REGISTERED PROTOCOL, VERBATIM (ITS QUESTION, DESIGN, DATA, EVALUATION AND DECISION RULES) · experiments/e3_pair/PROTOCOL-confirm.md · FIRST COMMIT 720cab6 2026-11-16 00:06 CET

Stage 4 — confirmatory proofs on the sealed C4 test split: protocol

Questions

Stage 1a (`PROTOCOL-ladder.md`, `runs/s1a/referee.md`) trained the matched 0.6B pair eight times longer than E3 and found, with one seed on the development split:

- both bases off the proof floor at their trained depth (ARLoop 4/32, PreNorm 3/32 at R = 3);
- with random-depth training, ARLoop better at R = 4 than at R = 1 (valid steps 61 against 55 of 64, all 6 discordant rows for R = 4, Holm p = 0.0625: not established), PreNorm not.

Stage 4 tests these leads once, on data never used: three seeds and the sealed C4 test split.

1. **P1, more loops help ARLoop (the 1a lead, `k.ladder-depth`).** Is ARLoop's random-depth adapter more often valid at R = 4 than at R = 1?
2. **P3, the pair at its trained depth (`CONFIRM-PROPOSAL.md` P1c).** Do ARLoop's native-depth adapters prove more test targets than PreNorm's at R = 3?

Design (as in Stage 1a unless listed here)

- **Bases, arms, recipe.** ARLoop-0.6B and PreNormLoop-0.6B at E3's revisions; arms native3 (every update at R = 3) and random124 (R drawn from {1, 2, 4} per update, latent's rule); E3's LoRA recipe with the cosine schedule to **4,096 updates** (Stage 1a's). No checkpoint selection: the adapter after update 4,096 is evaluated.
- **Seeds 11, 29, 47** (`CONFIRM-PROPOSAL.md`). Seed 11: the four Stage 1a adapters at 4,096 updates, reused unchanged (same recipe; their development results are already recorded). Seeds 29 and 47: eight new trainings with `train_lora.py --seed S --steps 4096 --label S4`. The seed sets the LoRA initialisation, the example order and the depth draws, identically for both bases.
- **Code.** E3's files and Stage 1a's ladder options, unchanged, plus:
  - `prep_split.py` (new): prepares a split with `data_prep.py`'s encoding and audit; the dev mode is a rehearsal.
  - `eval_proofs.py --split-dir` and `eval_steps.py --split-dir` (new options): the same rollout and the same single-step measure on a prepared split; without the option nothing changes.
  - `s4_gpu.sbatch`, `s4_cpu.sbatch`, `tools/submit_s4.sh`, `tools/launch_s4.sh`.

Data: the sealed C4 test split, opened once

- **Pins.** latent's C4 prepared corpus (`campaign4-hard-v1`, `preparation.json` SHA-256 `c48a8eec…`, admission `da141013…`). Its manifest pins the test selection `selection-test.json` (SHA-256 `248394bf9e64db355ecf8b8123f011231d992d8b50326557c98c65a00efa85bd`), drawn by latent with its selection plan `selection-plan.json` (SHA-256 `32938099b7ec5643d9c13373ac5fa15ca8f10005b45203e1896d6a361ff507f9`), the test rows (`36c05d9e…`, 12,675 rows, 206 sessions) and the test task metadata (`a4545be2…`). Only the selection plan and the manifest's counts and hashes were read before this protocol; no test row, target or roster entry.
- **Proof targets:** latent's own test targets, in the frozen order of `selection-test.json`: by its plan 48 targets over 8 strata (composition_holdout and depth_extrapolation × four difficulty strata).
- **Single-step roster (256 rows):** latent's own 128 test validation rows (16 per stratum, by its plan) **plus 128 rows drawn by a rule fixed now** (`prep_split.py`): per stratum (evaluation_regime, difficulty_stratum of the row's session), the eligible rows (not in latent's validation or generation rosters, not in a proof-target session), ordered by SHA-256 of `recur-s4-roster-v1|test|<row id>`, the first 16. If a stratum has fewer eligible rows, all are taken and the roster is smaller; the analysis uses the roster as drawn.
- **When.** `prep_split.py --split test` runs once, in a CPU job that starts only after all twelve adapters exist, their development evaluations (below) are written, and this protocol, `analyze_confirm.py` and the independent recount (`runs/s4/referee/s4_recount.py`, `runs/s4/referee/s4_replay.py`) are committed. It refuses to run if any

adapter is missing or incomplete, if a named development evaluation is missing, or if its output directory exists. Nothing is tuned on test; a failed or timed-out evaluation job is rerun with the same command (the evaluators resume), never changed.

Evaluation (E3's harness, unchanged: C4 budgets, greedy KV-cached decoding, 600 s per target, the pinned Python checker; not Lean)

- **Primary cells, per adapter:** random124 at R = 1 and R = 4, native3 at R = 3; each with autonomous proofs on the 48 test targets (`eval_proofs.py`) and single-step validity on the 256-row roster (`eval_steps.py`: one greedy action from the gold state; valid if the checker accepts the transition).
- **Secondary cells, as the queue allows:** random124 at R = 2 and R = 3, native3 at R = 4. Cells not evaluated by the analysis are reported as not measured.
- **Development record (before the test split is opened):** seeds 29 and 47 on the primary cells of the development split (32 targets, the 64-row roster), as in Stage 1a. Seed 11's development results are Stage 1a's.

Decision rules (fixed now)

Paired units: (seed, roster row) for single steps, (seed, target) for proofs, pooled over the three seeds. Exact McNemar tests are two-sided. The primary family is {P1, P3}, Holm-adjusted at 0.05.

- **P1, more loops help ARLoop.** ARLoop random124, valid single steps at R = 4 against R = 1, 768 paired units if the roster has 256 rows. **"More loops help ARLoop"** if the Holm-adjusted $p < 0.05$, the pooled difference is positive, and at least 2 of the 3 seeds have more rows valid only at R = 4 than only at R = 1. Otherwise "not established".
- **P3, the pair at R = 3.** native3 at R = 3, accepted test proofs ARLoop against PreNorm (144 paired units). **"ARLoop proves more"** if the Holm-adjusted $p < 0.05$, ARLoop has more targets proved only by it, and at least 2 of the 3 seeds favour ARLoop; "PreNorm proves more" symmetrically; otherwise "no detectable difference".
- **P2 (pre-registered secondary), the design contrast for depth.** Per paired unit d = (ARLoop valid at R4 − at R1) − (PreNorm valid at R4 − at R1); the mean of d over all units with a paired bootstrap that resamples roster rows within each seed (10,000 resamples, seed 20261115). **"ARLoop gains more from loops than PreNorm"** if the 95% interval excludes 0 and the mean is positive.
- **Further secondaries (reported, no claim without the primaries):**
  - S1: PreNorm's own depth effect, as P1 (McNemar p and per-seed signs).
  - S2: accepted proofs at R = 4 against R = 1, random124, each base (exact McNemar; described only if fewer than 5 targets are discordant).
  - S3 (`k.e3-rollouts`): accepted actions per attempt, ARLoop against PreNorm, paired by (seed, target), in the cells native3 at R = 4 and random124 at R = 1, 2, 4; exact sign test on the non-zero differences, Holm over the measured cells. Supported only if all four cells are measured and ARLoop's attempts are longer in each at Holm $p < 0.05$.

- S4: random124 against native3 at R = 4, each base (valid steps and proofs, exact McNemar).
- S5: per-seed tables for every measured cell; development against test for the primary cells.

- **Exploratory analyses** are labelled as such and logged with their commit times.

Amendment 1 (2026-11-16 00:15 CET, before any Stage 4 training or evaluation; nothing was measured): the trainings run with `train_lora.py --no-checkpoint`; `prep_split.py` stores the text fields and the sequence length of each row, not the token ids; `tools/launch_s4.sh` refuses to submit without 1.5 GB free under the quota, and `tools/submit_s4.sh` marks a code snapshot complete only after its files match the commit's hashes.

8.2 Results

THE RE-CHECK NOTE, VERBATIM · runs/s4/referee.md

Re-check note — Stage 4, confirmatory proofs on the sealed test split (2026-11-17 01:26 CET, at commit 12bbf1d)

Scope.

- Protocol `experiments/e3_pair/PROTOCOL-confirm.md`, with amendment 1 (registered before any Stage 4 run).
- Seeds 11, 29, 47 (seed 11: Stage 1a's adapters, reused unchanged); both bases; arms native3 and random124; the recipe of Stage 1a (4,096 LoRA updates).
- The test split of latent's Campaign 4, opened once: 48 proof targets (latent's own selection, SHA-256 `248394bf9e64…`) and a 224-row single-step roster (SHA-256 `ad36deed2744…`): latent's 128 validation rows plus 96 drawn by the registered rule (2 of the 8 strata had no eligible row, so the roster is smaller than the planned 256).
- Cells measured: native3@R3, random124@R1, random124@R4, native3@R4, random124@R2, random124@R3.
- 47 `rc-s4` jobs (`sacct.txt`): 45 completed with exit code 0, 2 ended otherwise (323384 rc-s4-smoke: FAILED, exit 1:0; 323436 rc-s4-test-secondary-arloop-native3-s47: TIMEOUT, exit 0:0); the analysis and the recount count only evaluation files they find complete; GPU time 69.6 GPU-h on `gpu_csi`, all jobs counted, 12.0 h of it job 323436, held until its time limit.
- The reading is generated by `experiments/e3_pair/analyze_confirm.py` (committed with the protocol, before any run): `s4.json`, `s4.md`. This note is written by `referee/write_note.py` from those files.

Method.

- `referee/s4_recount.py` shares no code with the analysis or the evaluators. From the raw per-target and per-row records it recounts accepted proofs, valid steps and accepted actions, and recomputes P1, P3 (with Holm), P2's mean, S1–S4 and the per-seed table.
- `referee/s4_replay.py` replays every recorded test attempt with the admitted checker and checks every target against latent's pinned test selection.

Result of the recount and the replay.

- 36 proof suites and as many step suites recounted: 0 disagreements with `s4.json`.
- P2's bootstrap interval, recomputed with another seed (7, 4,000 resamples): bounds differ from the analysis's by at most 0.15 points.
- 1,728 attempts replayed from 36 proof files: 34 kernel-checked QEDs, exactly the 34 accepted proofs; 0 problems.

The registered reading (from `s4.md`):

Test split: 48 proof targets, a 224-row single-step roster; seeds 11, 29, 47; measured cells: native3@R3, random124@R1, random124@R4, native3@R4, random124@R2, random124@R3.

P1 (ARLoop random124, valid steps at R = 4 against R = 1, 672 paired units): 64 only at R = 4, 1 only at R = 1 (+9.38 points); McNemar p = 3.58e-18, Holm p = 7.16e-18; seeds favouring R = 4: 3 of 3 — **more loops help ARLoop**.

P3 (native3 at R = 3, accepted test proofs, ARLoop against PreNorm, 144 paired units): 4 only ARLoop, 5 only PreNorm; McNemar p = 1, Holm p = 1; seeds favouring ARLoop 0, PreNorm 1 — **no detectable difference**.

P2 (design contrast for depth, 672 units): mean +4.32 points [+1.19, +7.44] — **ARLoop gains more from loops than PreNorm**.

S1 (PreNorm's own depth effect): 51 only at R = 4, 17 only at R = 1, McNemar p = 4.45e-05.

S2 (ARLoop, proofs at R = 4 against R = 1, random124): 4 / 1, McNemar p = 0.375 (tested).

S2 (PreNorm, proofs at R = 4 against R = 1, random124): 0 / 0, McNemar p = 1 (descriptive (fewer than 5 discordant targets)).

S3 (accepted actions per attempt, ARLoop against PreNorm): native3@R4: ARLoop longer 143, PreNorm longer 0, Holm p = 7.17e-43; random124@R1: ARLoop longer 78, PreNorm longer 38, Holm p = 0.000775; random124@R2: ARLoop longer 76, PreNorm longer 53, Holm p = 0.105; random124@R4: ARLoop longer 69, PreNorm longer 56, Holm p = 0.283 — supported: **no**.

S4 (ARLoop, random124 against native3 at R = 4): steps 37 / 8 (p = 1.54e-05); proofs 2 / 2 (p = 1).

S4 (PreNorm, random124 against native3 at R = 4): steps 122 / 107 (p = 0.355); proofs 0 / 0 (p = 1).

S5 (per seed): test accepted of 48 · valid of 224 · mean accepted actions; development (primary cells) accepted of 32 · valid of 64

BASE	ARM	R	SEED	TEST	DEVELOPMENT
ARLoop	native3	3	11	2 · 219 · 27.0	4 · 63
ARLoop	native3	3	29	3 · 218 · 28.1	3 · 62
ARLoop	native3	3	47	3 · 215 · 28.5	4 · 61
ARLoop	random124	1	11	0 · 198 · 21.0	0 · 55
ARLoop	random124	1	29	1 · 196 · 21.0	0 · 53
ARLoop	random124	1	47	0 · 200 · 20.0	0 · 54
ARLoop	random124	4	11	1 · 218 · 23.9	3 · 61
ARLoop	random124	4	29	2 · 220 · 23.8	4 · 62
ARLoop	random124	4	47	1 · 219 · 22.5	4 · 61
ARLoop	native3	4	11	0 · 205 · 24.9	–
ARLoop	native3	4	29	2 · 213 · 27.2	–
ARLoop	native3	4	47	2 · 210 · 26.0	–
ARLoop	random124	2	11	1 · 216 · 23.9	–
ARLoop	random124	2	29	1 · 215 · 23.5	–
ARLoop	random124	2	47	2 · 212 · 22.0	–
ARLoop	random124	3	11	0 · 219 · 24.1	–
ARLoop	random124	3	29	1 · 218 · 24.8	–
ARLoop	random124	3	47	3 · 218 · 22.6	–
PreNorm	native3	3	11	3 · 219 · 26.0	3 · 62
PreNorm	native3	3	29	3 · 216 · 24.7	2 · 63
PreNorm	native3	3	47	3 · 220 · 25.6	4 · 63
PreNorm	random124	1	11	0 · 166 · 22.0	0 · 49
PreNorm	random124	1	29	0 · 156 · 21.3	0 · 49
PreNorm	random124	1	47	0 · 162 · 17.4	0 · 49
PreNorm	random124	4	11	0 · 179 · 23.2	0 · 51
PreNorm	random124	4	29	0 · 169 · 23.4	0 · 51
PreNorm	random124	4	47	0 · 170 · 22.7	0 · 50
PreNorm	native3	4	11	0 · 166 · 7.3	–
PreNorm	native3	4	29	0 · 166 · 6.8	–
PreNorm	native3	4	47	0 · 171 · 5.5	–
PreNorm	random124	2	11	0 · 177 · 24.0	–
PreNorm	random124	2	29	0 · 167 · 21.3	–
PreNorm	random124	2	47	0 · 168 · 22.8	–

BASE	ARM	R	SEED	TEST	DEVELOPMENT
PreNorm	random124	3	11	0 · 177 · 23.6	–
PreNorm	random124	3	29	0 · 170 · 23.8	–
PreNorm	random124	3	47	0 · 171 · 22.7	–

Reading.

- **P1: more loops help ARLoop.** Its random-depth adapters are valid on 64 roster rows only at R = 4 against 1 only at R = 1, of 672 paired units (+9.38 points); Holm $p = 7.2 \times 10^{-18}$; 3 of 3 seeds favour R = 4. Stage 1a's lead is confirmed on the test split, with three seeds.
- **P3: no detectable difference** at the trained depth (R = 3): 4 test targets proved only by ARLoop against 5 only by PreNorm, of 144 paired units; Holm p = 1. With so few proofs a difference of a few targets is not excluded.
- **P2 (secondary), the design contrast for depth:** ARLoop's gain from R = 1 to R = 4 minus PreNorm's, +4.32 points [+1.19, +7.44] over 672 units: **ARLoop gains more from loops than PreNorm**.
- **S1, PreNorm's own depth effect:** 51 rows only at R = 4 against 17 only at R = 1 (McNemar $p = 4.5 \times 10^{-5}$; seeds favouring R = 4: 3 of 3).
- **S2, proofs at R = 4 against R = 1 (random-depth adapters):** ARLoop-0.6B 4 against 1 (McNemar p = 0.38); PreNormLoop-0.6B 0 against 0 (McNemar p = 1; described only: fewer than 5 discordant targets).
- **S3, how far the attempts get (`k.e3-rollouts`):** native3@R4: ARLoop longer on 143, PreNorm on 0 (Holm $p = 7.2 \times 10^{-43}$); random124@R1: ARLoop longer on 78, PreNorm on 38 (Holm $p = 7.7 \times 10^{-4}$); random124@R2: ARLoop longer on 76, PreNorm on 53 (Holm p = 0.1); random124@R4: ARLoop longer on 69, PreNorm on 56 (Holm p = 0.28). Supported: **no**.
- **S4, random-depth against native-depth adapters at R = 4:** ARLoop-0.6B steps 37 against 8, proofs 2 against 2; PreNormLoop-0.6B steps 122 against 107, proofs 0 against 0.

What it means for the question "does more training make looping pay off?"

- With random-depth training, more loops help ARLoop on unseen proof goals (its single steps are valid more often at R = 4 than at R = 1): sense 1 of a result of looping (`CAMPAIGN.md`), registered and confirmed with three seeds. ARLoop also gains more from loops than PreNorm (P2). PreNorm's random-depth adapters gain from more loops too (S1, secondary: 51 against 17 rows, McNemar $p = 4.5 \times 10^{-5}$): the gain is not ARLoop's alone, but larger in it.

Limits.

- Three seeds of one recipe, one model size and one corpus; 48 test targets and a 224-row roster.
- A null result bounds what three seeds can detect; it is not evidence that no effect exists.

- The test split of Campaign 4 is now used: it cannot serve another confirmatory test of these questions.

Addendum (the coordinator, 2026-11-17 01:18 CET): how two kinds of job ended, from their logs and the test files' modification times (`runs/s4/logs-extract.txt`) and `sacct.txt`.

- Job 323436 (the secondary tests of ARLoop native3, seed 47) wrote its last line, "=== done", at 13:17:09 on 2026-11-16, then stayed allocated until its 12-hour limit (TIMEOUT at 00:56:26 on 2026-11-17). WMI's SLURM controller did not answer the login node from 13:28 (the coordinator's `squeue`) until shortly before the second runs below began (14:51). Its four suites are complete, each with its summary, and are counted.
- Jobs 323438, 323439, 323441 and 323442 wrote "=== done" between 13:29:15 and 14:00:49; 323442's log adds "REQUEST_COMPLETE_BATCH_SCRIPT failed: Connection reset by peer" (14:17:06). All four then ran their commands a second time, between 14:51 and 14:53, found every suite complete and ended within seconds; `sacct.txt` shows the second run's elapsed time. No test file changed after 14:00:48, before the second runs: the counted files are those of the first runs.

Re-checked by the coordinator with independent code; human check pending.

9 STAGE 2 · TRAINING AT DEPTH

Train the whole model at four loops, *then compare*

Stage 2 asks the between-models question: does a model trained, in all its parameters, at four loops beat the same model trained at three, on the same data and budget, and does ARLoop gain more than the PreNorm loop? The plan's Stage 3 (both designs trained from scratch at 140M, a token ladder, planned for after the talk on Helios's GH200 partition) has not run: that partition is down, as the Stage 2 protocol records.

9.1 Design

REGISTERED PROTOCOL, VERBATIM (ITS QUESTION, DESIGN AND DECISION RULES) · `experiments/s2_depth/PROTOCOL.md` · FIRST COMMIT `86c6f82` 2026-11-16 00:05 CET

Stage 2 — continued training of the whole 0.6B pair at depth (`k.depth-training`): protocol

Question

Stage 1c trained only the routing of a fourth loop of ARLoop-0.6B and found no gain on GSM8K; the frozen read-out kept routing around loop 4. Does training the **whole** model at R = 4 make a fourth loop useful, and more so for ARLoop than for the PreNorm loop? (Sense 2 of `CAMPAIGN.md`: a model trained at more loops against one trained at fewer, on the same data and the same number of updates.)

Design

- **Models:** ARLoop-0.6B and PreNormLoop-0.6B at the revisions of `experiments/e1_release/loopctl.py`.
- **Arms** (identical recipe, data, order and seed; only the loop count differs):
  - `R3`: every update at R = 3 (the control: the released depth);
  - `R4`: every update at R = 4. ARLoop's 36 routing queries and 36 norm weights of loop 4 start as independent copies of loop 1's (E1's `cycle` rule; `depthmodel.py`) and are trained with everything else;
  - `Rrand` (registered, conditional): R drawn per update from {2, 3, 4, 5} with `random.Random(11)`, the same draws for both bases; ARLoop's table extended to 5 loops by the same rule.
- **Recipe** (`train_depth.py`, `RECIPE`): every parameter trained; the model computes in BF16 as released (BF16 weights, BF16 autocast, FP32 routing) with FP32 master weights and FP32 gradient accumulation; AdamW (0.9, 0.95, eps 1e-8, no weight decay), gradient clipping at global norm 1.0; peak rate 3e-5 after 100 linear warm-up updates, cosine to 3e-6 at the last update; 16 sequences of 2,048 tokens per update, one pass over the shard in the order `random.Random(11).shuffle(...)`; mean next-token cross-entropy; seed 11; deterministic algorithms, TF32 off. Micro-batch 2 on both GPUs (the gradient of an update is the same sum).
- **Hardware assignment:** each base's two arms on one GPU, one after the other, so that each primary comparison is within one GPU type: ARLoop on the L40S, PreNorm on the RTX 5090 (or the reverse, if memory requires it; recorded). R4 first, then R3.
- **Evaluation** (`eval_depth.py`: E1's `run_lmeval.py` line for line, with the trained weights loaded): GSM8K (gsm8k_cot, 5-shot, BOS, flexible extract, all 1,319 test items, greedy, <= 1,024 new tokens) and WikiText-2 bits per byte (0-shot, no BOS). Primary evaluations first: the R4-trained model at R = 4 and the R3-trained model at R = 3. Then the cross-depth evaluations (R4-trained at R = 3, R3-trained at R = 4) and WikiText at both depths. The released models' E1 runs (`runs/e1/gsm8k-bos/{ar06,pn06}-R3`, `ar06-R4-cycle`, `pn06-R4`) are the untrained references.

Decision rules (fixed now)

1. **Primary**, per base: the R4-trained model at R = 4 against the R3-trained model at R = 3, paired over the 1,319 GSM8K items: difference in accuracy, exact two-sided McNemar on the discordant items, paired bootstrap 95% interval (seed 20261115, 10,000 resamples; E1's `paired`). Holm over the two bases. "Training at depth makes the fourth loop pay off" for a base if its difference is positive and its Holm-adjusted p is below 0.05.
2. **Design contrast** (secondary): (ARLoop's R4-at-4 minus R3-at-3) minus (PreNorm's), paired bootstrap 95% interval over items (the same resamples for both bases' differences). "ARLoop gains more from depth training" if the interval excludes 0 on the positive side.

3. **Training effect** (secondary): each trained model against its released model at the same R (E1's samples); the R3 arm against the released R = 3 isolates what 0.1B tokens of math text do without a depth change.
4. **Cross-depth** (secondary, descriptive with paired statistics): R4-trained at R = 3 and R3-trained at R = 4.
5. **WikiText-2** bits per byte at R = 3 and R = 4 per model (descriptive).
6. **Random depth** (only if `Rrand` runs; sense 1): accuracy by R in {2, 3, 4, 5} within each random-depth model, R = 5 against R = 3 with E1's paired statistics; descriptive.
7. Exploratory analyses are labelled as such and logged with their commit times.

Registered deviations from the campaign plan, before any run: Platform: WMI; Design reduced; Data: Stage 1c's audited FineMath-4+ shard; Token budget rule.

Amendment 1 (2026-11-16 00:17 CET, before any main run; only gates and speed measurements have run): Micro-batch 1 on both GPUs; Node-local storage; Gates in the main jobs; Budget: the whole shard, U = 3,051 updates.

Amendment 2 (2026-11-16 11:41 CET, before the random-depth arm runs; no Stage 2 result has been read): The condition of `Rrand` is met for PreNorm; ARLoop's `Rrand` does not run; The job; Rule 6 unchanged; Expectations (not rules).

9.2 Results

THE RE-CHECK NOTE, VERBATIM · `runs/s2/referee.md`

Re-check note — Stage 2, continued whole-model training at depth (2026-11-17 01:19 CET, at commit 12bbf1d)

Scope.

- Protocol `experiments/s2_depth/PROTOCOL.md`, with amendments 1 (registered before any main run) and 2 (before the random-depth arm ran: it runs for PreNorm only).
- Both released 0.6B models, every parameter trained on Stage 1c's FineMath-4+ shard: 3,051 updates, 0.10B tokens, one seed; arms R3 and R4 for both (R3 is the control at the released depth); for PreNorm only, also Rrand, with R drawn per update from {2, 3, 4, 5}.
- GPUs: ARLoop-0.6B on NVIDIA L40S, PreNormLoop-0.6B on NVIDIA GeForce RTX 5090; each base's comparison stays on one GPU type. Training time 23.7 h in total (receipts).
- 15 `rc-s2` jobs (`sacct.txt`): 8 completed with exit code 0, 7 ended otherwise (323382 rc-s2-gate-ar: FAILED, exit 1:0; 323383 rc-s2-gate-pn: FAILED, exit 1:0; 323385 rc-s2-gate-ar: FAILED, exit 1:0; 323386 rc-s2-gate-pn: FAILED, exit 1:0; 323396 rc-s2-ar: CANCELLED, exit 0:0; 323397 rc-s2-pn: FAILED, exit 1:0; 323398 rc-s2-ar: FAILED, exit 1:0); the analysis and the recount count only evaluation files they find complete.
- The reading is generated by `experiments/s2_depth/analyze_depth.py` (committed before any result): `analysis/depth.json`, `analysis/depth.md`. This note is written by `referee/write_note.py` from those files.

Method. `referee/s2_recount.py` shares no code with the analysis. It re-scores every GSM8K item of every Stage 2 evaluation and of E1's four released references from the raw generations (E1's re-implementation of the flexible-extract rule), checks each against the recorded score, and recomputes rule 1 with Holm, rule 2's point estimate and rules 3, 4 and 6.

Result of the recount. 13 comparisons and 15 runs re-scored: 0 disagreements. Bootstrap intervals, recomputed with another seed (7, 4,000 resamples), differ from the analysis's by at most 0.08 points (rules 1 and 2).

The registered reading (from `analysis/depth.md`):

GSM8K (1,319 items, 5-shot CoT, BOS, flexible-extract): arloop-R3-R3 34.34%, arloop-R4-R3 35.63%, arloop-R4-R4 35.25%, prenorm-R3-R3 30.17%, prenorm-R3-R4 16.53%, prenorm-R4-R3 17.59%, prenorm-R4-R4 30.78%, prenorm-Rrand-R2 6.97%, prenorm-Rrand-R3 24.26%, prenorm-Rrand-R4 25.32%, prenorm-Rrand-R5 24.03%; released: arloop-R3 38.97%, arloop-R4 38.82%, prenorm-R3 33.81%, prenorm-R4 16.15%.

Rule 1 (primary; R4-trained at R = 4 against R3-trained at R = 3; Holm over the two bases):

- ARLoop-0.6B: +0.91 points [−0.61, +2.43], discordant 57/45, McNemar p = 0.276, Holm p = 0.552; passes: **no**.
- PreNormLoop-0.6B: +0.61 points [−1.29, +2.58], discordant 90/82, McNemar p = 0.594, Holm p = 0.594; passes: **no**.

Rule 2 (design contrast, ARLoop's difference minus PreNorm's): +0.30 points [−2.20, +2.73]; ARLoop gains more: **no**.

Rule 3 (trained against released at the same R):

- arloop-R3-R3 vs released R3: −4.62 points [−6.82, −2.43], discordant 78/139, McNemar p = 4.16e-05.
- arloop-R4-R4 vs released R4: −3.56 points [−5.84, −1.36], discordant 94/141, McNemar p = 0.00262.
- prenorm-R3-R3 vs released R3: −3.64 points [−5.69, −1.52], discordant 75/123, McNemar p = 0.000794.
- prenorm-R4-R4 vs released R4: +14.63 points [+11.98, +17.29], discordant 276/83, McNemar p = 2.44e-25.

Rule 4 (cross-depth):

- arloop-R4-R3 vs arloop-R3-R3: +1.29 points [−0.53, +3.11], discordant 86/69, McNemar p = 0.199.
- prenorm-R4-R3 vs prenorm-R3-R3: −12.59 points [−15.09, −10.08], discordant 70/236, McNemar p = 3.45e-22.
- prenorm-R3-R4 vs prenorm-R3-R3: −13.65 points [−16.22, −10.99], discordant 79/259, McNemar p = 1.72e-23.

Rule 5 (WikiText-2 bits per byte): prenorm-R3-R3 0.7524, prenorm-R3-R4 0.9973, prenorm-R4-R3 0.9924, prenorm-R4-R4 0.7603; released: arloop-R3 0.7247, arloop-R4 0.7320, prenorm-R3 0.7378, prenorm-R4 0.9798.

Rule 6 (random depth):

- prenorm-Rrand-R2 vs prenorm-Rrand-R3: −17.29 points [−19.64, −15.01], discordant 31/259, McNemar p = 5.68e-46.
- prenorm-Rrand-R4 vs prenorm-Rrand-R3: +1.06 points [−0.76, +2.81], discordant 80/66, McNemar p = 0.282.
- prenorm-Rrand-R5 vs prenorm-Rrand-R3: −0.23 points [−2.35, +1.90], discordant 97/100, McNemar p = 0.887.

Not evaluated (secondary): gsm8k arloop-R3-R4, wikitext arloop-R3-R3, wikitext arloop-R3-R4, wikitext arloop-R4-R3, wikitext arloop-R4-R4.

Reading.

- **ARLoop-0.6B: not established.** Trained and run at R = 4 against trained and run at R = 3: +0.91 points [−0.61, +2.43] (57 against 45 discordant items), Holm p = 0.55. A gain of up to 2.43 points is not excluded.
- **PreNormLoop-0.6B: not established.** Trained and run at R = 4 against trained and run at R = 3: +0.61 points [−1.29, +2.58] (90 against 82 discordant items), Holm p = 0.59. A gain of up to 2.58 points is not excluded.
- **Design contrast (secondary):** ARLoop's difference minus PreNorm's, +0.30 points [−2.20, +2.73]: **not established**.
- **Training effect, ARLoop-0.6B:** the control trained at R = 3 against the released model at R = 3, −4.62 points [−6.82, −2.43] (McNemar $p = 4.2 \times 10^{-5}$): what 0.1B tokens of math text do without a change of depth.
- **Training effect, PreNormLoop-0.6B:** the control trained at R = 3 against the released model at R = 3, −3.64 points [−5.69, −1.52] (McNemar $p = 7.9 \times 10^{-4}$): what 0.1B tokens of math text do without a change of depth.
- **Cross-depth (descriptive):** arloop-R4-R3 against arloop-R3-R3 +1.29 points [−0.53, +3.11]; prenorm-R4-R3 against prenorm-R3-R3 −12.59 points [−15.09, −10.08]; prenorm-R3-R4 against prenorm-R3-R3 −13.65 points [−16.22, −10.99].
- **WikiText-2 bits per byte (descriptive):** prenorm-R3-R3 0.7524, prenorm-R3-R4 0.9973, prenorm-R4-R3 0.9924, prenorm-R4-R4 0.7603; released: arloop-R3 0.7247, arloop-R4 0.7320, prenorm-R3 0.7378, prenorm-R4 0.9798.
- **Random depth, PreNorm only (rule 6, descriptive; amendment 2):** PreNormLoop-0.6B trained the same way, but with R drawn per update from {2, 3, 4, 5}. GSM8K accuracy at R = 2, 3, 4 and 5: 6.97%, 24.26%, 25.32%, 24.03%. R = 5 against R = 3: −0.23 points [−2.35, +1.90] (97 items solved only at R = 5, 100 only at R = 3; McNemar p = 0.89). Against R = 3 also: R = 2 −17.29 points [−19.64, −15.01]; R = 4 +1.06 points [−0.76, +2.81]. No decision rule and no multiplicity correction: a description, not a test. ARLoop's random-depth arm did not run (amendment 2).

- **Not evaluated (secondary):** gsm8k arloop-R3-R4, wikitext arloop-R3-R3, wikitext arloop-R3-R4, wikitext arloop-R4-R3, wikitext arloop-R4-R4.

What it means for the question "does more training make looping pay off?"

- Training the whole model at R = 4 does not make the fourth loop pay off on GSM8K for either base, at this budget: no gain is detected. One seed and 0.1B tokens do not exclude a gain with longer training.
- The extra training itself costs accuracy at the released depth: both controls trained at R = 3 are below the released models at R = 3 (ARLoop-0.6B −4.62 points [−6.82, −2.43]; PreNormLoop-0.6B −3.64 points [−5.69, −1.52]). The depth comparison is made after a training that lowers GSM8K accuracy at this budget.
- Within PreNorm's random-depth model (descriptive, sense 1), accuracy is not detectably different between R = 5 and R = 3 (−0.23 points [−2.35, +1.90]). A description of one model and one seed, not a registered test; it says nothing about ARLoop, whose random-depth arm did not run.

Limits.

- One seed; 0.1B tokens of math text (Stage 1c's shard, not the authors' pretraining mixture); one model size.
- The two bases trained on different GPU types (numerics only; each base's own comparison is within one type).
- A null result bounds what this budget can detect; it is not evidence that no effect exists.
- The random-depth arm ran for PreNorm only (amendment 2) and is read descriptively (rule 6): it says nothing about ARLoop.

Addendum (the coordinator, 2026-11-17 01:18 CET): why ARLoop's job 323398 ended FAILED, from its log (`runs/s2/logs-extract.txt`) and the job list of `tools/launch_s2.sh`.

- After its two trainings it ran the GSM8K evaluations in the listed order: trained at R = 4, run at R = 4 (from 14:24); trained at R = 3, run at R = 3 (from 16:15); trained at R = 4, run at R = 3 (from 17:31). All three are complete, the two primary ones included.
- The fourth, the model trained at R = 3 run at R = 4 (rule 4, secondary), stopped at once at 18:59: "ARLoop's routing table has 108 rows: it cannot run 4 loops". A model trained at R = 3 has routing queries for three loops only; running it at R = 4 needs a rule for the missing queries, such as E1's `cycle`, and the evaluation applied none. The job stops at its first failure, so ARLoop's four WikiText-2 evaluations (rule 5, descriptive) did not run either: these are the five evaluations listed as not evaluated.
- The six earlier jobs that ended otherwise (323382–323397) are gate and launch attempts before the main runs, fixed before them (`runs/s2/sacct-early.txt`, amendment 1).

Re-checked by the coordinator with independent code; human check pending.

11 findings *about the paper and its code*

What we found in reading the paper and its released code, published with the digest on 2026-11-15 (F1–F9) and 2026-11-16 (F10 and F11) (corrected on 2026-11-15, 2026-11-16: see the entries). Severity: wording (the text says more or less than shown), scope (a claim rests on narrower evidence than it seems), implicit design (a design fact the paper leaves implicit), context, or confirmed. None of them changes a number in the paper's tables.

F1 · WORDING / IMPLICIT DESIGN

The loop is weight-tied only in its transformer blocks: routing has one query **and one RMSNorm per effective sublayer** (+ final), 334,848 parameters in the 0.6B model, exactly the parameter gap to PreNormLoop. The paper's equation (§3.1, eq. 2) shows only the per-sublayer queries; its appendix (A.1, on the controlled 140M/370M models; App. B on the released models is silent) adds that the normalization parameters are per effective sublayer too, with a separate final aggregation, but never counts them [corrected on 2026-11-16: this entry first said that the paper does not mention the norms]. Consequence: the released ARLoop is undefined for more loops than trained without a rule; PreNormLoop is not.

Place: §3.2; HF `modeling_arloop.py` l. 112–116 · **Evidence:** config + code + HF parameter counts

F2 · CONTEXT

The stage-3 mixture (from 80% of training) includes Dolmino `math/gsm8k` (weight 1e-4) and GSM-style synthetic sets. Same for both 0.6B models, so the pair comparison is fair; comparisons with Huginn/Qwen mix architecture with data. Dolmino's GSM8K data is GSM8K's training split, not its test split (OLMo 2 report, arXiv:2501.00656, §4.2.1: "we include the training split of GSM8K").

Place: §5; `configs/mixed_data/0p6b_arloop_r3_180b.yaml` · **Evidence:** config lines ≈195–240; `scripts/data/sources/dolmino.py`; OLMo 2 report §4.2.1

F3 · WORDING

"Huginn-3.5B (the only fully-open looped LLM as of our release date)" is too strong: Parcae released looped LMs of 140M–1.3B (`SandyResearch/parcae-*`, created 2026-04-08; code `sandyresearch/parcae`, MIT; trained on FineWeb-Edu) and McLeish et al. released a retrofitted recurrent OLMo-2 1B (`smcleish/Recurrent-OLMo-2-0425-train-recurrence-32`, 2025-11-08, Apache-2.0, open data). Parcae is even a baseline in ARLoop's Tab. 1. The claim holds only if "LLM" means a looped model pretrained from scratch on a realistic data mixture at ≥ 1B parameters: Parcae's models are trained on FineWeb-Edu only, and McLeish et al.'s model is retrofitted from a pretrained OLMo-2.

Place: Abstract · **Evidence:** HF API `createdAt`/licence, GitHub API, 2026-11-15

F4 · SCOPE

The extrapolation conclusion rests on per-sequence Poisson full-history 140M/370M models (trained with R ≤ 4) whose queries for the unseen depths R = 5–8 are newly initialised and trained on 0.5B tokens with the rest frozen, not on the released models.

Place: §6 vs App. D.2 · **Evidence:** App. D.2 text

F5 · CONFIRMED (WORKS)

`ARLoop-baseline-0.6B` and `ARLoop-1.3B` redirect to `PreNormLoop-0.6B` and `ARLoop-1.2B`.

Place: paper links · **Evidence:** HTTP 307 targets

F6 · WORDING

The first residual block is initialised with the embedding (`blocks=[h0]`, `partial=h0`), so $b_1 = h_0 + \sum_{j=1}^{S} z_j$, not $\sum z_j$ as in eq. (4); the embedding is a source twice until b_1 leaves the window: it is last read by sublayer 54 (the attention sublayer of effective layer 27) and absent from sublayer 55 on [code + exp: smoke routing probe, job 322097; corrected on 2026-11-15 after a simulation of the window schedule]. Same in training and HF code.

Place: eq. (4) vs code · **Evidence:** `loopedlm/models/arloop/fla_softmax.py` l. 198–199; HF `modeling_arloop.py` l. 189

F7 · WORDING

"Larger gains at deeper recurrences": the gain over PreNormLoop at 140M is −0.53 PPL at R = 3 but −0.50 at R = 5, so not monotone (holds against R = 1, and at 370M from R = 1 to 3).

Place: §4.1, Tab. 1 · **Evidence:** Tab. 1 values (LaTeX)

F8 · WORDING / IMPLICIT DESIGN

The first routing call (sublayer 0) has nothing to read but the embedding: two copies of h_0 in the code (F6), so its weights are 0.5/0.5 for any query and its output is h_0 (under eq. (4) it would read h_0 alone, with weight 1, equally independent of the query). Either way the query and norm of sublayer 0 (3,072 parameters) are inert and receive no gradient. Measured: exactly 0.5/0.5 at call 0 on the smoke probe. With all queries zero and no window, the routing weights are uniform, so the routed state is PreNorm's state divided by the number of sources and, after the norm, the sublayer inputs are exactly PreNorm's (eqs. 2–4); the code's double-counted h_0 breaks this identity.

Place: code · **Evidence:** HF `modeling_arloop.py` l. 189–195; `runs/e1/smoke/routing-ar/routing.json`

F9 · SCOPE (ROBUSTNESS)

The GSM8K gap of the pair depends on one evaluation detail. With BOS prepended (the authors' default, which we follow): +5.16 points [+2.65, +7.66], p = 6.5 × 10^{-5}. Without BOS: +2.20 points [−0.45, +4.85], p = 0.12. BOS lowers PreNormLoop-0.6B (−2.43 points [−4.02, −0.76], p = 0.0052) and has no detectable effect on ARLoop-0.6B (+0.53 points [−1.06, +2.12], p = 0.56). The paper evaluates every model the same way, so its table is consistent, but the gap's size is not robust to this choice; no BOS sensitivity is reported.

Place: §5, Tab. 3; evaluation script · **Evidence:** E1.1 and the BOS control, independently re-checked (`runs/e1/referee.md`)

F10 · IMPLICIT DESIGN

Each sublayer's routing query q and routing-norm weight g act only through their product $e = q \odot g$, and in the released checkpoints $|q| = |g|$ coordinate by coordinate (ARLoop-0.6B: exactly on 161,791 of 165,888 coordinates, otherwise within 5.1 × 10^{-12}). The magnitudes are bimodal, and on 99,053 coordinates (59.7%) the product underflows to exactly 0 in float32: the routing scores use, on average, 40% of the hidden dimensions (loop 1: 27%, loop 3: 49%). ARLoop-1.2B is alike (67.6% zero). The paper's equation has one query per sublayer and its appendix names per-sublayer normalization parameters, but it says neither that the two act only as a product nor that the product is mostly zero.

Place: §3.1 eq. (2), App. A.1; HF `modeling_arloop.py` l. 139; checkpoints · **Evidence:** safetensors read directly; independently re-checked (`runs/f10/referee.md`)

F11 · SCOPE

The main text presents the sliding window as enough for loops (C2) and never mentions value residuals; the appendix shows the window's edge depends on them. App. A.1: "removing value residuals substantially degrades our sliding-window depth-attention variant"; Tab. 7 (140M, R = 3, without value residuals): window 18.20, PreNorm 18.19, full history 17.52. Both 0.6B models in Tab. 3 use value residuals, so the pair comparison is unaffected.

Place: §4.1 vs App. A.1, A.3 (Tab. 7) · **Evidence:** App. A.1 l. 19, A.3 Tab. 7; main text searched

11 DISCUSSION

What decides whether *an extra loop is used*

The study set out to ask what decides whether a looped model uses an extra loop. The evidence, by strength:

ESTABLISHED (T4, RE-CHECKED)

The trained depth bounds what the models do

Every released model collapses below its trained depth (0% at one and two loops) and no gain is detected above it, under either inference rule and at both sizes (E1.3). In the canary, trained with random depths, no depth gain is detected even after 30 times the training (E2, Stage 1b). The proof models were at the floor after 512 LoRA updates and left it after 4,096 (E3, Stage 1a).

EXPLORATORY (NO REGISTERED RULE); THE PROBE IS DESCRIPTIVE AND NOT RE-CHECKED

One loop past the trained depth, ARLoop loses less

From R = 3 to R = 4, ARLoop-0.6B loses 10.39 points under repeat and 0.15 under cycle, PreNormLoop-0.6B 17.66. Under cycle the routing probe is consistent with routing around the extra loop: the read-out keeps 93.4% of its weight on loop 3's last block. Under repeat it does not, and ARLoop still loses 10.39 points. Training that loop's routing does not change the picture (Stage 1c). This fits the hunch that attention over depth makes an extra loop cheap to ignore rather than useful; the paper's own App. D.2 points the same way at smaller scale.

CONFIRMED (T4, RE-CHECKED, THREE SEEDS)

Random-depth training lets ARLoop use depth on proofs

The one sign of a result of looping came from proofs: after 4,096 updates with a random number of loops, ARLoop-0.6B's adapter improves with more loops on proofs, valid steps and cross-entropy, while PreNormLoop-0.6B's does not (Stage 1a, one seed); the registered test misses after Holm correction. Stage 4 repeated the training with three seeds and opened the sealed test split once: ARLoop's random-depth adapters make 64 single steps valid only at R = 4 against 1 only at R = 1, of 672 paired rows (Holm $p = 7.2 \times 10^{-18}$; 3 of 3 seeds favour R = 4): more loops help ARLoop, as registered. ARLoop's gain from one to four loops minus PreNorm's (secondary): +4.32 points [+1.19, +7.44], ARLoop gains more. PreNorm's random-depth adapters gain from more loops too (S1, secondary: 51 against 17 single steps, McNemar $p = 4.5 \times 10^{-5}$), less than ARLoop's: the effect is not ARLoop's alone.

TRAINING AT DEPTH (T4, RE-CHECKED, ONE SEED)

The whole model trained at four loops

Every parameter of both 0.6B models trained further on 0.1B tokens of math text (3,051 updates), at R = 3 and at R = 4: no gain is detected for ARLoop-0.6B and PreNormLoop-0.6B (ARLoop-0.6B +0.91 points [−0.61, +2.43], Holm p = 0.55; PreNormLoop-0.6B +0.61 points [−1.29, +2.58], Holm p = 0.59). ARLoop's difference minus PreNorm's (secondary): +0.30 points [−2.20, +2.73], not established. The training itself costs accuracy at R = 3, against the released models: ARLoop-0.6B −4.62 points [−6.82, −2.43]; PreNormLoop-0.6B −3.64 points [−5.69, −1.52]. PreNorm was also trained with a random number of loops per update, 2 to 5 (rule 6, descriptive; ARLoop's random-depth arm did not run): GSM8K accuracy 6.97%, 24.26%, 25.32%, 24.03% at R = 2, 3, 4, 5; R = 5 against R = 3, −0.23 points [−2.35, +1.90].

A MEASUREMENT CAVEAT

The headline gap depends on BOS

The paper's GSM8K gap for the matched pair is reproduced with the authors' settings (+5.16 points) but is inconclusive without BOS (+2.20, an interval containing 0), because BOS lowers PreNormLoop-0.6B and has no detectable effect on ARLoop-0.6B (finding F9). Claims about the residual design that rest on one benchmark number should carry this sensitivity.

Interpretation (ours, not a registered result)

On these tasks the amount of training at the tested depth matters first: no comparison here ranks it against the residual design, but no case without such training shows a detectable use of an extra loop. The residual design shows its effects in two places: at the trained depth, where the matched pair's GSM8K gap is reproduced but its size depends on BOS, and past the trained depth, where ARLoop loses less (exploratory). A third place appears with random-depth training: ARLoop turns more loops into valid proof steps on unseen goals and gains more from them than the PreNorm loop (Stage 4).

12 LIMITATIONS

Threats to validity, *stated plainly*

Across the study

- All results are tier T4, computational evidence on samples, mostly with one seed (E3, Stage 1a, Stage 1c, Stage 2) and small item counts (32 proof targets, a 64-row step roster).
- The decision rules are per experiment. Holm's correction is applied within the primary families of Stage 1a, Stage 1c, Stage 4 and Stage 2, not across the study; E1's cells are read one by one; E2 uses its candidate-effect rule without a multiplicity correction and reports every comparison.
- Exploratory analyses (E1's cross-model comparisons at $R \geq 4$ and its routing probe at $R = 4$, E3's rollout diagnostics) were defined after results were seen; they are labelled and carry no decision rule.

E1

- The Hugging Face backend instead of vLLM (which the released README names for the paper's numbers); absolute accuracies are about two points lower (the cause was not tested), the paired gap matches the paper's; one run per cell (a repeated run gave identical generations).
- $R > 3$ uses inference-only rules for ARLoop's missing routing parameters (queries and norms), not the paper's App. D.2 procedure, which trains newly initialised queries for the unseen depths; Stage 1c adds trained routing.
- The routing probe is descriptive, on 64 prompts. MATH, MBPP, HumanEval and the layer-effect claim (C4) were not re-run.

E2 and Stage 1b

- One toy task with 93 training examples and tiny models: a model can memorise it without using depth, so a null here says little about language models.
- No arm has value residuals [16] (registered), which the paper's App. A.1 and A.3 report its sliding window needs; the ARLoop arms test ARLoop's routing, not the full released design.

E3 and Stage 1a

- Development split only (the test split was kept sealed for Stage 4); 32 targets and a 64-row roster.
- Declared deviations from Campaign 4: KV-cached greedy decoding (measured on the untuned bases only), 600 s per target instead of 1,200 s per 32-target suite, and no checkpoint selection; no BOS, following latent's data contract.

- Stage 1a's 1,024 and 2,048-update rungs are snapshots of one 4,096-update schedule, not separately scheduled runs.
- ARLoop's random-depth arm trains extra routing parameters for the added loop; PreNorm has none to train.

Stage 1c

- Routing only, with the read-out frozen by design; 0.1B tokens, one seed, one model.

Not run

- The paper's controlled 140M/370M studies (C1, C2, C5), which cost about 12k GPU-h; the throughput claim (C7).

13 REPRODUCIBILITY

Every number *traceable to a file*

Everything needed to check a number in this report is in the repository: the protocols, the code that ran (snapshotted per commit), the receipts, the small result files, the re-check scripts and their outputs. Large per-item logs stay on the cluster; their SHA-256 hashes are kept next to the results.

TABLE 20 The job ledger in brief (every job is recorded in the files named; GPUs: RTX PRO 6000, and RTX 4090 for E2 and Stage 1b).

PART	JOBS	COMPUTE	RESULTS
E1	10 jobs (322097, 322118, 322150, 322183, 322184, 322185, 322186, 322256, 322258, 322259): evaluations, routing probes, the smoke test and the BOS check	17.5 GPU-h of GSM8K generation (receipts; one GPU per run)	`runs/e1/summary.json`, `runs/e1/referee.md`
E2	3 jobs, smoke tests included	0.12 GPU-h (sacct)	`experiments/e2_small/results/322153/`, `runs/e2/referee.md`
E3	9 jobs	4.35 GPU-h (sacct)	`experiments/e3_pair/results/`, `runs/e3/referee.md`
Stage 1a	16 jobs	30.2 GPU-h (sacct)	`runs/s1a/`
Stage 1b	2 jobs (322560, 322667)	1.9 GPU-h of training and evaluation (summaries)	`experiments/e2_small/results/322560/`, `…/322667/`, `runs/s1b/`
Stage 1c	25 jobs	8.5 GPU-h of training and 3.1 GPU-h of evaluation (receipts)	`runs/s1c/`

Re-running a step

Table 21 gives the command line of every analysis and re-check script: as its docstring states it, or, where it states none, as the script reads its inputs (fixed paths or an environment variable). They run on a CPU node next to the raw files, most with the standard library only (the E1 analysis also needs matplotlib, and the Stage

1a replay the admitted checker, `peano_lab`).

TABLE 21 Analysis and re-check entry points (from each script's docstring or its way of reading its inputs).

SCRIPT	COMMAND
experiments/e1_release/analyze.py	python3 -I experiments/e1_release/analyze.py (on WMI)
runs/e1/referee/e1_referee.py	python3 -I runs/e1/referee/e1_referee.py > runs/e1/referee/e1_referee.json
experiments/e2_small/scripts/analyze.py	python3 analyze.py RUN_DIR [--reference LATENT_RUN_DIR] [--out OUT_DIR]
experiments/e3_pair/analyze.py	E3_ROOT=… python3 -I experiments/e3_pair/analyze.py (on WMI)
runs/e3/referee/e3_referee.py	python3 e3_referee.py > referee.json
experiments/e1_release/analyze_loop4.py	python3 -I analyze_loop4.py E1_RUNS S1C_ROOT OUT_DIR
experiments/e3_pair/analyze_ladder.py	python3 -I analyze_ladder.py E3_EVALS S1A_EVALS OUT_DIR
runs/s1a/referee/s1a_recount.py	python3 -I s1a_recount.py E3_EVALS S1A_EVALS LADDER_JSON OUT.json
runs/s1a/referee/s1a_replay.py	python3 -I s1a_replay.py PEANO_ROOT DATA_DIR E3_EVALS S1A_EVALS OUT.json
runs/s1b/referee/s1b_recount.py	python3 -I s1b_recount.py RUNS_ROOT ANALYSIS_ROOT OUT.json
runs/s1c/referee/s1c_recount.py	python3 -I s1c_recount.py E1_RUNS S1C_ROOT LOOP4_JSON OUT.json

This report

`python3 -I report/build_report.py --pdf` rebuilds the HTML and the PDF from the files listed in the colophon; the build stops if any guarded sentence no longer matches its files.

Every paired comparison

The tables below are complete: every paired comparison computed by the registered analyses, including those that no rule reads. Differences are in points; intervals are paired bootstrap intervals.

A.1 E1: every paired comparison

TABLE 22 E1, all paired comparisons of `runs/e1/summary.json` (exploratory ones marked as such).

COMPARISON (A VS B)	N	A	B	A - B	95% INTERVAL	DISCORDANT	MCNEMAR P
ARLoop-0.6B vs PreNormLoop-0.6B at R = 3	1,319	38.97%	33.81%	+5.16	[+2.65, +7.66]	176 / 108	6.5×10^{-5}
ar06-R3: BOS vs no BOS	1,319	38.97%	38.44%	+0.53	[−1.06, +2.12]	56 / 49	0.56
pn06-R3: BOS vs no BOS	1,319	33.81%	36.24%	−2.43	[−4.02, −0.76]	46 / 78	0.0052
no BOS: ARLoop-0.6B vs PreNormLoop-0.6B at R = 3	1,319	38.44%	36.24%	+2.20	[−0.45, +4.85]	176 / 147	0.12
ar06-R1 vs ar06-R3	1,319	0.00%	38.97%	−38.97	[−41.62, −36.32]	0 / 514	3.7×10^{-155}
ar06-R2 vs ar06-R3	1,319	0.00%	38.97%	−38.97	[−41.62, −36.32]	0 / 514	3.7×10^{-155}
ar06-R4-cycle vs ar06-R3	1,319	38.82%	38.97%	−0.15	[−1.97, +1.67]	75 / 77	0.94
ar06-R4-repeat vs ar06-R3	1,319	28.58%	38.97%	−10.39	[−12.96, −7.81]	90 / 227	8.4×10^{-15}
ar06-R5-repeat vs ar06-R3	1,319	2.81%	38.97%	−36.16	[−38.97, −33.36]	22 / 499	1.0×10^{-118}
ar06-R6-repeat vs ar06-R3	1,319	1.74%	38.97%	−37.23	[−40.03, −34.42]	17 / 508	7.1×10^{-127}
pn06-R1 vs pn06-R3	1,319	0.00%	33.81%	−33.81	[−36.39, −31.31]	0 / 446	1.1×10^{-134}
pn06-R2 vs pn06-R3	1,319	0.00%	33.81%	−33.81	[−36.39, −31.31]	0 / 446	1.1×10^{-134}
pn06-R4 vs pn06-R3	1,319	16.15%	33.81%	−17.66	[−20.39, −14.94]	73 / 306	4.9×10^{-35}
pn06-R5 vs pn06-R3	1,319	1.67%	33.81%	−32.15	[−34.72, −29.57]	11 / 435	3.5×10^{-113}
pn06-R6 vs pn06-R3	1,319	0.61%	33.81%	−33.21	[−35.86, −30.63]	7 / 445	1.3×10^{-121}
ar12-R2 vs ar12-R3	1,319	0.00%	63.00%	−63.00	[−65.66, −60.35]	0 / 831	1.4×10^{-250}
ar12-R4-repeat vs ar12-R3	1,319	59.21%	63.00%	−3.79	[−6.07, −1.52]	100 / 150	0.0019
exploratory: ar06-R4-repeat vs pn06-R4	1,319	28.58%	16.15%	+12.43	[+9.78, +15.16]	253 / 89	2.4×10^{-19}
exploratory: ar06-R4-cycle vs pn06-R4	1,319	38.82%	16.15%	+22.67	[+19.86, +25.55]	369 / 70	4.0×10^{-50}
exploratory: ar06-R5-repeat vs pn06-R5	1,319	2.81%	1.67%	+1.14	[+0.08, +2.27]	36 / 21	0.063
exploratory: ar06-R6-repeat vs pn06-R6	1,319	1.74%	0.61%	+1.14	[+0.38, +1.97]	22 / 7	0.0081

A.2 E2 (1,000 updates): every comparison

TABLE 23 E2, job 322153: all comparisons of `analysis.json`, by family; per-seed count differences for seeds 11, 29, 47.

X	Y	R	SPLIT	N	X - Y	95% INTERVAL	PER SEED	DISCORDANT	MCNEMAR P	CANDIDATE
PRIMARY (6)										
prenorm-loop	latent-recurrent	4	iid	105	−6.67	[−13.33, +0.00]	−5 / +0 / −2	4 / 11	0.12	no
prenorm-loop	latent-recurrent	16	iid	105	−5.71	[−13.33, +0.95]	−4 / −1 / −1	5 / 11	0.21	no
arloop-untied	latent-recurrent	4	iid	105	−3.81	[−11.43, +3.81]	−5 / +1 / +0	7 / 11	0.48	no
arloop-untied	latent-recurrent	16	iid	105	−5.71	[−13.33, +0.95]	−5 / +1 / −2	5 / 11	0.21	no
arloop-tied	latent-recurrent	4	iid	105	−4.76	[−12.38, +2.86]	−5 / +0 / +0	7 / 12	0.36	no
arloop-tied	latent-recurrent	16	iid	105	−4.76	[−12.38, +2.86]	−5 / +0 / +0	7 / 12	0.36	no
SECONDARY: OTHER LOOPS AND SPLITS (39)										
prenorm-loop	latent-recurrent	1	iid	105	−5.71	[−12.38, +0.95]	−3 / −4 / +1	4 / 10	0.18	no
prenorm-loop	latent-recurrent	1	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
prenorm-loop	latent-recurrent	1	ood_operand	96	−9.38	[−15.62, −4.17]	−3 / −2 / −4	0 / 9	0.0039	**yes**
prenorm-loop	latent-recurrent	2	iid	105	−4.76	[−12.38, +2.86]	−4 / +0 / −1	6 / 11	0.33	no
prenorm-loop	latent-recurrent	2	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no

X	Y	R	SPLIT	N	X - Y	95% INTERVAL	PER SEED	DISCORDANT	MCNEMAR P	CANDIDATE
prenorm-loop	latent-recurrent	2	ood_operand	96	−3.12	[−8.33, +1.04]	+0 / +0 / −3	1 / 4	0.375	no
prenorm-loop	latent-recurrent	4	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
prenorm-loop	latent-recurrent	4	ood_operand	96	−2.08	[−7.29, +3.12]	+1 / +0 / −3	2 / 4	0.6875	no
prenorm-loop	latent-recurrent	8	iid	105	−7.62	[−15.24, −0.95]	−5 / −1 / −2	4 / 12	0.077	**yes**
prenorm-loop	latent-recurrent	8	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
prenorm-loop	latent-recurrent	8	ood_operand	96	−3.12	[−8.33, +2.08]	−1 / +0 / −2	2 / 5	0.45	no
prenorm-loop	latent-recurrent	16	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
prenorm-loop	latent-recurrent	16	ood_operand	96	−3.12	[−8.33, +2.08]	−1 / +0 / −2	2 / 5	0.45	no
arloop-untied	latent-recurrent	1	iid	105	−5.71	[−12.38, +0.95]	−5 / −2 / +1	4 / 10	0.18	no
arloop-untied	latent-recurrent	1	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-untied	latent-recurrent	1	ood_operand	96	−6.25	[−12.50, −1.04]	−3 / −1 / −2	1 / 7	0.07	**yes**
arloop-untied	latent-recurrent	2	iid	105	−3.81	[−11.43, +3.81]	−5 / +1 / +0	7 / 11	0.48	no
arloop-untied	latent-recurrent	2	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-untied	latent-recurrent	2	ood_operand	96	−3.12	[−8.33, +2.08]	−2 / +1 / −2	2 / 5	0.45	no
arloop-untied	latent-recurrent	4	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-untied	latent-recurrent	4	ood_operand	96	−3.12	[−8.33, +2.08]	−2 / +2 / −3	2 / 5	0.45	no
arloop-untied	latent-recurrent	8	iid	105	−5.71	[−13.33, +0.95]	−6 / +1 / −1	5 / 11	0.21	no
arloop-untied	latent-recurrent	8	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-untied	latent-recurrent	8	ood_operand	96	−3.12	[−8.33, +2.08]	−2 / +2 / −3	2 / 5	0.45	no
arloop-untied	latent-recurrent	16	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-untied	latent-recurrent	16	ood_operand	96	−3.12	[−8.33, +2.08]	−2 / +2 / −3	2 / 5	0.45	no
arloop-tied	latent-recurrent	1	iid	105	−5.71	[−12.38, +0.95]	−3 / −4 / +1	4 / 10	0.18	no
arloop-tied	latent-recurrent	1	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-tied	latent-recurrent	1	ood_operand	96	−7.29	[−14.58, −1.04]	−3 / +0 / −4	2 / 9	0.065	no
arloop-tied	latent-recurrent	2	iid	105	−6.67	[−14.29, +0.95]	−5 / −1 / −1	5 / 12	0.14	no
arloop-tied	latent-recurrent	2	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-tied	latent-recurrent	2	ood_operand	96	−3.12	[−8.33, +2.08]	−2 / +2 / −3	2 / 5	0.45	no
arloop-tied	latent-recurrent	4	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-tied	latent-recurrent	4	ood_operand	96	−3.12	[−8.33, +2.08]	−2 / +2 / −3	2 / 5	0.45	no
arloop-tied	latent-recurrent	8	iid	105	−4.76	[−12.38, +2.86]	−5 / +0 / +0	7 / 12	0.36	no
arloop-tied	latent-recurrent	8	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-tied	latent-recurrent	8	ood_operand	96	−3.12	[−8.33, +2.08]	−2 / +2 / −3	2 / 5	0.45	no
arloop-tied	latent-recurrent	16	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-tied	latent-recurrent	16	ood_operand	96	−3.12	[−8.33, +2.08]	−2 / +2 / −3	2 / 5	0.45	no

SECONDARY: AGAINST THE DEPTH-MATCHED CONTROL (63)

latent-recurrent	standard-depth	1	iid	105	+3.81	[−2.86, +10.48]	+0 / +4 / +0	9 / 5	0.42	no
latent-recurrent	standard-depth	1	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
latent-recurrent	standard-depth	1	ood_operand	96	+8.33	[+2.08, +14.58]	+3 / +1 / +4	9 / 1	0.021	**yes**
latent-recurrent	standard-depth	2	iid	105	+3.81	[−3.81, +11.43]	+1 / +1 / +2	10 / 6	0.45	no
latent-recurrent	standard-depth	2	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
latent-recurrent	standard-depth	2	ood_operand	96	+4.17	[+0.00, +9.38]	+2 / −1 / +3	5 / 1	0.22	no
latent-recurrent	standard-depth	4	iid	105	+3.81	[−3.81, +11.43]	+1 / +1 / +2	10 / 6	0.45	no
latent-recurrent	standard-depth	4	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
latent-recurrent	standard-depth	4	ood_operand	96	+4.17	[+0.00, +9.38]	+2 / −1 / +3	5 / 1	0.22	no
latent-recurrent	standard-depth	8	iid	105	+3.81	[−3.81, +11.43]	+1 / +1 / +2	10 / 6	0.45	no
latent-recurrent	standard-depth	8	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no

X	Y	R	SPLIT	N	X - Y	95% INTERVAL	PER SEED	DISCORDANT	MCNEMAR P	CANDIDATE
latent-recurrent	standard-depth	8	ood_operand	96	+4.17	[+0.00, +9.38]	+2 / −1 / +3	5 / 1	0.22	no
latent-recurrent	standard-depth	16	iid	105	+3.81	[−3.81, +11.43]	+1 / +1 / +2	10 / 6	0.45	no
latent-recurrent	standard-depth	16	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
latent-recurrent	standard-depth	16	ood_operand	96	+4.17	[+0.00, +9.38]	+2 / −1 / +3	5 / 1	0.22	no
prenorm-loop	standard-depth	1	iid	105	−1.90	[−8.57, +4.76]	−3 / +0 / +1	5 / 7	0.77	no
prenorm-loop	standard-depth	1	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
prenorm-loop	standard-depth	1	ood_operand	96	−1.04	[−3.12, +0.00]	+0 / −1 / +0	0 / 1	1	no
prenorm-loop	standard-depth	2	iid	105	−0.95	[−6.67, +4.76]	−3 / +1 / +1	5 / 6	1	no
prenorm-loop	standard-depth	2	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
prenorm-loop	standard-depth	2	ood_operand	96	+1.04	[−2.08, +4.17]	+2 / −1 / +0	2 / 1	1	no
prenorm-loop	standard-depth	4	iid	105	−2.86	[−8.57, +2.86]	−4 / +1 / +0	4 / 7	0.55	no
prenorm-loop	standard-depth	4	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
prenorm-loop	standard-depth	4	ood_operand	96	+2.08	[−2.08, +6.25]	+3 / −1 / +0	3 / 1	0.625	no
prenorm-loop	standard-depth	8	iid	105	−3.81	[−10.48, +1.90]	−4 / +0 / +0	4 / 8	0.39	no
prenorm-loop	standard-depth	8	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
prenorm-loop	standard-depth	8	ood_operand	96	+1.04	[−2.08, +5.21]	+1 / −1 / +1	2 / 1	1	no
prenorm-loop	standard-depth	16	iid	105	−1.90	[−8.57, +4.76]	−3 / +0 / +1	6 / 8	0.79	no
prenorm-loop	standard-depth	16	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
prenorm-loop	standard-depth	16	ood_operand	96	+1.04	[−2.08, +5.21]	+1 / −1 / +1	2 / 1	1	no
arloop-untied	standard-depth	1	iid	105	−1.90	[−8.57, +4.76]	−5 / +2 / +1	6 / 8	0.79	no
arloop-untied	standard-depth	1	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-untied	standard-depth	1	ood_operand	96	+2.08	[−2.08, +6.25]	+0 / +0 / +2	3 / 1	0.625	no
arloop-untied	standard-depth	2	iid	105	+0.00	[−6.67, +6.67]	−4 / +2 / +2	7 / 7	1	no
arloop-untied	standard-depth	2	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-untied	standard-depth	2	ood_operand	96	+1.04	[−2.08, +5.21]	+0 / +0 / +1	2 / 1	1	no
arloop-untied	standard-depth	4	iid	105	+0.00	[−6.67, +6.67]	−4 / +2 / +2	7 / 7	1	no
arloop-untied	standard-depth	4	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-untied	standard-depth	4	ood_operand	96	+1.04	[−2.08, +5.21]	+0 / +1 / +0	2 / 1	1	no
arloop-untied	standard-depth	8	iid	105	−1.90	[−8.57, +4.76]	−5 / +2 / +1	6 / 8	0.79	no
arloop-untied	standard-depth	8	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-untied	standard-depth	8	ood_operand	96	+1.04	[−2.08, +5.21]	+0 / +1 / +0	2 / 1	1	no
arloop-untied	standard-depth	16	iid	105	−1.90	[−8.57, +4.76]	−4 / +2 / +0	6 / 8	0.79	no
arloop-untied	standard-depth	16	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-untied	standard-depth	16	ood_operand	96	+1.04	[−2.08, +5.21]	+0 / +1 / +0	2 / 1	1	no
arloop-tied	standard-depth	1	iid	105	−1.90	[−8.57, +4.76]	−3 / +0 / +1	5 / 7	0.77	no
arloop-tied	standard-depth	1	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-tied	standard-depth	1	ood_operand	96	+1.04	[−2.08, +5.21]	+0 / +1 / +0	2 / 1	1	no
arloop-tied	standard-depth	2	iid	105	−2.86	[−9.52, +3.81]	−4 / +0 / +1	5 / 8	0.58	no
arloop-tied	standard-depth	2	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-tied	standard-depth	2	ood_operand	96	+1.04	[−2.08, +5.21]	+0 / +1 / +0	2 / 1	1	no
arloop-tied	standard-depth	4	iid	105	−0.95	[−7.62, +5.71]	−4 / +1 / +2	7 / 8	1	no
arloop-tied	standard-depth	4	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-tied	standard-depth	4	ood_operand	96	+1.04	[−2.08, +5.21]	+0 / +1 / +0	2 / 1	1	no
arloop-tied	standard-depth	8	iid	105	−0.95	[−7.62, +5.71]	−4 / +1 / +2	7 / 8	1	no
arloop-tied	standard-depth	8	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-tied	standard-depth	8	ood_operand	96	+1.04	[−2.08, +5.21]	+0 / +1 / +0	2 / 1	1	no
arloop-tied	standard-depth	16	iid	105	−0.95	[−7.62, +5.71]	−4 / +1 / +2	7 / 8	1	no
arloop-tied	standard-depth	16	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-tied	standard-depth	16	ood_operand	96	+1.04	[−2.08, +5.21]	+0 / +1 / +0	2 / 1	1	no
standard-parameter	standard-depth	–	iid	105	+0.95	[−5.71, +7.62]	−3 / +2 / +2	7 / 6	1	no
standard-parameter	standard-depth	–	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no

X	Y	R	SPLIT	N	X − Y	95% INTERVAL	PER SEED	DISCORDANT	MCNEMAR P	CANDIDATE
standard-parameter	standard-depth	–	ood_operand	96	−1.04	[−3.12, +0.00]	+0 / −1 / +0	0 / 1	1	no
SECONDARY: THE DEEPEST AGAINST THE TRAINING MEAN, WITHIN AN ARM (12)										
latent-recurrent	latent-recurrent	16 vs 4	iid	105	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
latent-recurrent	latent-recurrent	16 vs 4	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
latent-recurrent	latent-recurrent	16 vs 4	ood_operand	96	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
prenorm-loop	prenorm-loop	16 vs 4	iid	105	+0.95	[−2.86, +4.76]	+1 / −1 / +1	3 / 2	1	no
prenorm-loop	prenorm-loop	16 vs 4	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
prenorm-loop	prenorm-loop	16 vs 4	ood_operand	96	−1.04	[−4.17, +2.08]	−2 / +0 / +1	1 / 2	1	no
arloop-untied	arloop-untied	16 vs 4	iid	105	−1.90	[−6.67, +2.86]	+0 / +0 / −2	2 / 4	0.6875	no
arloop-untied	arloop-untied	16 vs 4	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-untied	arloop-untied	16 vs 4	ood_operand	96	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-tied	arloop-tied	16 vs 4	iid	105	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-tied	arloop-tied	16 vs 4	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-tied	arloop-tied	16 vs 4	ood_operand	96	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no

A.3 Stage 1b (10,000 updates): every comparison

TABLE 24 Stage 1b, job 322560 (10,000 updates): all comparisons of `analysis.json`, by family; per-seed count differences for seeds 11, 29, 47.

X	Y	R	SPLIT	N	X − Y	95% INTERVAL	PER SEED	DISCORDANT	MCNEMAR P	CANDIDATE
PRIMARY (6)										
prenorm-loop	latent-recurrent	4	iid	105	−4.76	[−13.33, +3.81]	−6 / −2 / +3	9 / 14	0.4	no
prenorm-loop	latent-recurrent	16	iid	105	−4.76	[−14.29, +3.81]	−6 / −3 / +4	10 / 15	0.42	no
arloop-untied	latent-recurrent	4	iid	105	−1.90	[−10.48, +5.71]	−6 / +1 / +3	9 / 11	0.82	no
arloop-untied	latent-recurrent	16	iid	105	−4.76	[−13.33, +3.81]	−6 / +0 / +1	8 / 13	0.38	no
arloop-tied	latent-recurrent	4	iid	105	−0.95	[−9.52, +7.62]	−4 / +1 / +2	10 / 11	1	no
arloop-tied	latent-recurrent	16	iid	105	−1.90	[−10.48, +6.67]	−4 / +1 / +1	10 / 12	0.83	no
SECONDARY: OTHER LOOPS AND SPLITS (39)										
prenorm-loop	latent-recurrent	1	iid	105	−3.81	[−12.38, +4.76]	−5 / −2 / +3	10 / 14	0.54	no
prenorm-loop	latent-recurrent	1	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
prenorm-loop	latent-recurrent	1	ood_operand	96	−9.38	[−16.67, −3.12]	−7 / −1 / −1	2 / 11	0.022	**yes**
prenorm-loop	latent-recurrent	2	iid	105	−1.90	[−10.48, +6.67]	−5 / −2 / +5	11 / 13	0.84	no
prenorm-loop	latent-recurrent	2	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
prenorm-loop	latent-recurrent	2	ood_operand	96	−10.42	[−17.71, −4.17]	−6 / −1 / −3	1 / 11	0.0063	**yes**
prenorm-loop	latent-recurrent	4	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
prenorm-loop	latent-recurrent	4	ood_operand	96	−11.46	[−18.75, −5.21]	−7 / −1 / −3	1 / 12	0.0034	**yes**
prenorm-loop	latent-recurrent	8	iid	105	−3.81	[−12.38, +4.76]	−6 / −2 / +4	10 / 14	0.54	no
prenorm-loop	latent-recurrent	8	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
prenorm-loop	latent-recurrent	8	ood_operand	96	−11.46	[−18.75, −5.21]	−7 / −1 / −3	1 / 12	0.0034	**yes**
prenorm-loop	latent-recurrent	16	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
prenorm-loop	latent-recurrent	16	ood_operand	96	−12.50	[−19.79, −6.25]	−7 / −2 / −3	0 / 12	4.9×10^{-4}	**yes**
arloop-untied	latent-recurrent	1	iid	105	−3.81	[−12.38, +4.76]	−6 / +0 / +2	8 / 12	0.5	no
arloop-untied	latent-recurrent	1	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-untied	latent-recurrent	1	ood_operand	96	−8.33	[−14.58, −2.08]	−7 / +0 / −1	2 / 10	0.039	no
arloop-untied	latent-recurrent	2	iid	105	+0.00	[−8.57, +8.57]	−5 / +2 / +3	11 / 11	1	no
arloop-untied	latent-recurrent	2	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-untied	latent-recurrent	2	ood_operand	96	−8.33	[−15.62, −2.08]	−6 / +0 / −2	2 / 10	0.039	no
arloop-untied	latent-recurrent	4	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no

X	Y	R	SPLIT	N	X - Y	95% INTERVAL	PER SEED	DISCORDANT	MCNEMAR P	CANDIDATE
arloop-untied	latent-recurrent	4	ood_operand	96	−8.33	[−15.62, −1.04]	−7 / +1 / −2	3 / 11	0.057	no
arloop-untied	latent-recurrent	8	iid	105	−5.71	[−13.33, +1.90]	−6 / +1 / −1	6 / 12	0.24	no
arloop-untied	latent-recurrent	8	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-untied	latent-recurrent	8	ood_operand	96	−9.38	[−16.67, −3.12]	−7 / +0 / −2	2 / 11	0.022	no
arloop-untied	latent-recurrent	16	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-untied	latent-recurrent	16	ood_operand	96	−8.33	[−15.62, −1.04]	−7 / +1 / −2	3 / 11	0.057	no
arloop-tied	latent-recurrent	1	iid	105	−2.86	[−10.48, +4.76]	−3 / −1 / +1	7 / 10	0.63	no
arloop-tied	latent-recurrent	1	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-tied	latent-recurrent	1	ood_operand	96	−7.29	[−14.58, +0.00]	−7 / +2 / −2	4 / 11	0.12	no
arloop-tied	latent-recurrent	2	iid	105	+0.00	[−7.62, +7.62]	−3 / +1 / +2	9 / 9	1	no
arloop-tied	latent-recurrent	2	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-tied	latent-recurrent	2	ood_operand	96	−7.29	[−14.58, +0.00]	−6 / +2 / −3	4 / 11	0.12	no
arloop-tied	latent-recurrent	4	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-tied	latent-recurrent	4	ood_operand	96	−8.33	[−16.67, −1.04]	−7 / +2 / −3	4 / 12	0.077	no
arloop-tied	latent-recurrent	8	iid	105	−1.90	[−10.48, +6.67]	−4 / +1 / +1	10 / 12	0.83	no
arloop-tied	latent-recurrent	8	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-tied	latent-recurrent	8	ood_operand	96	−10.42	[−17.71, −3.12]	−7 / +0 / −3	2 / 12	0.013	no
arloop-tied	latent-recurrent	16	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-tied	latent-recurrent	16	ood_operand	96	−10.42	[−17.71, −3.12]	−7 / +0 / −3	2 / 12	0.013	no

SECONDARY: AGAINST THE DEPTH-MATCHED CONTROL (63)

latent-recurrent	standard-depth	1	iid	105	+2.86	[−6.67, +12.38]	−1 / +0 / +4	14 / 11	0.69	no
latent-recurrent	standard-depth	1	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
latent-recurrent	standard-depth	1	ood_operand	96	+11.46	[+5.21, +17.71]	+7 / +2 / +2	11 / 0	9.8×10^{-4}	**yes**
latent-recurrent	standard-depth	2	iid	105	+1.90	[−6.67, +10.48]	−1 / +0 / +3	12 / 10	0.83	no
latent-recurrent	standard-depth	2	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
latent-recurrent	standard-depth	2	ood_operand	96	+11.46	[+5.21, +17.71]	+6 / +2 / +3	11 / 0	9.8×10^{-4}	**yes**
latent-recurrent	standard-depth	4	iid	105	+2.86	[−5.71, +11.43]	−1 / +0 / +4	13 / 10	0.68	no
latent-recurrent	standard-depth	4	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
latent-recurrent	standard-depth	4	ood_operand	96	+12.50	[+6.25, +19.79]	+7 / +2 / +3	12 / 0	4.9×10^{-4}	**yes**
latent-recurrent	standard-depth	8	iid	105	+2.86	[−5.71, +11.43]	−1 / +0 / +4	13 / 10	0.68	no
latent-recurrent	standard-depth	8	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
latent-recurrent	standard-depth	8	ood_operand	96	+12.50	[+6.25, +19.79]	+7 / +2 / +3	12 / 0	4.9×10^{-4}	**yes**
latent-recurrent	standard-depth	16	iid	105	+2.86	[−5.71, +11.43]	−1 / +0 / +4	13 / 10	0.68	no
latent-recurrent	standard-depth	16	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
latent-recurrent	standard-depth	16	ood_operand	96	+12.50	[+6.25, +19.79]	+7 / +2 / +3	12 / 0	4.9×10^{-4}	**yes**
prenorm-loop	standard-depth	1	iid	105	−0.95	[−8.57, +6.67]	−6 / −2 / +7	9 / 10	1	no
prenorm-loop	standard-depth	1	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
prenorm-loop	standard-depth	1	ood_operand	96	+2.08	[+0.00, +5.21]	+0 / +1 / +1	2 / 0	0.5	no
prenorm-loop	standard-depth	2	iid	105	+0.00	[−7.62, +7.62]	−6 / −2 / +8	10 / 10	1	no
prenorm-loop	standard-depth	2	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
prenorm-loop	standard-depth	2	ood_operand	96	+1.04	[+0.00, +3.12]	+0 / +1 / +0	1 / 0	1	no
prenorm-loop	standard-depth	4	iid	105	−1.90	[−10.48, +6.67]	−7 / −2 / +7	11 / 13	0.84	no
prenorm-loop	standard-depth	4	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no

X	Y	R	SPLIT	N	X - Y	95% INTERVAL	PER SEED	DISCORDANT	MCNEMAR P	CANDIDATE
prenorm-loop	standard-depth	4	ood_operand	96	+1.04	[+0.00, +3.12]	+0 / +1 / +0	1 / 0	1	no
prenorm-loop	standard-depth	8	iid	105	−0.95	[−9.52, +7.62]	−7 / −2 / +8	11 / 12	1	no
prenorm-loop	standard-depth	8	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
prenorm-loop	standard-depth	8	ood_operand	96	+1.04	[+0.00, +3.12]	+0 / +1 / +0	1 / 0	1	no
prenorm-loop	standard-depth	16	iid	105	−1.90	[−10.48, +6.67]	−7 / −3 / +8	11 / 13	0.84	no
prenorm-loop	standard-depth	16	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
prenorm-loop	standard-depth	16	ood_operand	96	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-untied	standard-depth	1	iid	105	−0.95	[−9.52, +6.67]	−7 / +0 / +6	10 / 11	1	no
arloop-untied	standard-depth	1	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-untied	standard-depth	1	ood_operand	96	+3.12	[+0.00, +7.29]	+0 / +2 / +1	3 / 0	0.25	no
arloop-untied	standard-depth	2	iid	105	+1.90	[−6.67, +10.48]	−6 / +2 / +6	12 / 10	0.83	no
arloop-untied	standard-depth	2	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-untied	standard-depth	2	ood_operand	96	+3.12	[+0.00, +7.29]	+0 / +2 / +1	3 / 0	0.25	no
arloop-untied	standard-depth	4	iid	105	+0.95	[−6.67, +8.57]	−7 / +1 / +7	11 / 10	1	no
arloop-untied	standard-depth	4	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-untied	standard-depth	4	ood_operand	96	+4.17	[+1.04, +8.33]	+0 / +3 / +1	4 / 0	0.125	no
arloop-untied	standard-depth	8	iid	105	−2.86	[−10.48, +4.76]	−7 / +1 / +3	8 / 11	0.65	no
arloop-untied	standard-depth	8	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-untied	standard-depth	8	ood_operand	96	+3.12	[+0.00, +7.29]	+0 / +2 / +1	3 / 0	0.25	no
arloop-untied	standard-depth	16	iid	105	−1.90	[−9.52, +5.71]	−7 / +0 / +5	9 / 11	0.82	no
arloop-untied	standard-depth	16	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-untied	standard-depth	16	ood_operand	96	+4.17	[+1.04, +8.33]	+0 / +3 / +1	4 / 0	0.125	no
arloop-tied	standard-depth	1	iid	105	+0.00	[−8.57, +7.62]	−4 / −1 / +5	10 / 10	1	no
arloop-tied	standard-depth	1	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-tied	standard-depth	1	ood_operand	96	+4.17	[+1.04, +8.33]	+0 / +4 / +0	4 / 0	0.125	no
arloop-tied	standard-depth	2	iid	105	+1.90	[−6.67, +10.48]	−4 / +1 / +5	12 / 10	0.83	no
arloop-tied	standard-depth	2	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-tied	standard-depth	2	ood_operand	96	+4.17	[+1.04, +8.33]	+0 / +4 / +0	4 / 0	0.125	no
arloop-tied	standard-depth	4	iid	105	+1.90	[−6.67, +10.48]	−5 / +1 / +6	13 / 11	0.84	no
arloop-tied	standard-depth	4	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-tied	standard-depth	4	ood_operand	96	+4.17	[+1.04, +8.33]	+0 / +4 / +0	4 / 0	0.125	no
arloop-tied	standard-depth	8	iid	105	+0.95	[−7.62, +9.52]	−5 / +1 / +5	12 / 11	1	no
arloop-tied	standard-depth	8	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-tied	standard-depth	8	ood_operand	96	+2.08	[+0.00, +5.21]	+0 / +2 / +0	2 / 0	0.5	no
arloop-tied	standard-depth	16	iid	105	+0.95	[−7.62, +9.52]	−5 / +1 / +5	12 / 11	1	no
arloop-tied	standard-depth	16	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-tied	standard-depth	16	ood_operand	96	+2.08	[+0.00, +5.21]	+0 / +2 / +0	2 / 0	0.5	no
standard-parameter	standard-depth	–	iid	105	−2.86	[−10.48, +4.76]	−4 / +0 / +1	7 / 10	0.63	no
standard-parameter	standard-depth	–	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
standard-parameter	standard-depth	–	ood_operand	96	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no

SECONDARY: THE DEEPEST AGAINST THE TRAINING MEAN, WITHIN AN ARM (12)

latent-recurrent	latent-recurrent	16 vs 4	iid	105	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
latent-recurrent	latent-recurrent	16 vs 4	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
latent-recurrent	latent-recurrent	16 vs 4	ood_operand	96	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
prenorm-loop	prenorm-loop	16 vs 4	iid	105	+0.00	[−4.76, +4.76]	+0 / −1 / +1	3 / 3	1	no
prenorm-loop	prenorm-loop	16 vs 4	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
prenorm-loop	prenorm-loop	16 vs 4	ood_operand	96	−1.04	[−3.12, +0.00]	+0 / −1 / +0	0 / 1	1	no
arloop-untied	arloop-untied	16 vs 4	iid	105	−2.86	[−6.67, +0.00]	+0 / −1 / −2	0 / 3	0.25	no
arloop-untied	arloop-untied	16 vs 4	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-untied	arloop-untied	16 vs 4	ood_operand	96	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no

X	Y	R	SPLIT	N	X - Y	95% INTERVAL	PER SEED	DISCORDANT	MCNEMAR P	CANDIDATE
arloop-tied	arloop-tied	16 vs 4	iid	105	−0.95	[−2.86, +0.00]	+0 / +0 / −1	0 / 1	1	no
arloop-tied	arloop-tied	16 vs 4	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-tied	arloop-tied	16 vs 4	ood_operand	96	−2.08	[−5.21, +0.00]	+0 / −2 / +0	0 / 2	0.5	no

A.4 Stage 1b (30,000 updates): every comparison

TABLE 25 Stage 1b, job 322667 (30,000 updates): all comparisons of `analysis.json`, by family; per-seed count differences for seeds 11, 29, 47.

X	Y	R	SPLIT	N	X - Y	95% INTERVAL	PER SEED	DISCORDANT	MCNEMAR P	CANDIDATE
PRIMARY (6)										
prenorm-loop	latent-recurrent	4	iid	105	−5.71	[−15.24, +2.86]	−8 / −1 / +3	10 / 16	0.33	no
prenorm-loop	latent-recurrent	16	iid	105	−6.67	[−16.19, +2.86]	−9 / +2 / +0	10 / 17	0.25	no
arloop-untied	latent-recurrent	4	iid	105	−6.67	[−16.19, +1.90]	−6 / −2 / +1	9 / 16	0.23	no
arloop-untied	latent-recurrent	16	iid	105	−7.62	[−17.14, +1.90]	−6 / −1 / −1	9 / 17	0.17	no
arloop-tied	latent-recurrent	4	iid	105	−6.67	[−16.19, +1.90]	−10 / +0 / +3	10 / 17	0.25	no
arloop-tied	latent-recurrent	16	iid	105	−4.76	[−13.33, +3.81]	−10 / +1 / +4	10 / 15	0.42	no
SECONDARY: OTHER LOOPS AND SPLITS (39)										
prenorm-loop	latent-recurrent	1	iid	105	−6.67	[−14.29, +0.95]	−9 / −2 / +4	7 / 14	0.19	no
prenorm-loop	latent-recurrent	1	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
prenorm-loop	latent-recurrent	1	ood_operand	96	−1.04	[−8.33, +6.25]	−4 / +1 / +2	7 / 8	1	no
prenorm-loop	latent-recurrent	2	iid	105	−4.76	[−13.33, +3.81]	−8 / +0 / +3	10 / 15	0.42	no
prenorm-loop	latent-recurrent	2	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
prenorm-loop	latent-recurrent	2	ood_operand	96	−2.08	[−9.38, +5.21]	−4 / +2 / +0	6 / 8	0.79	no
prenorm-loop	latent-recurrent	4	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
prenorm-loop	latent-recurrent	4	ood_operand	96	−3.12	[−10.42, +4.17]	−4 / +2 / −1	5 / 8	0.58	no
prenorm-loop	latent-recurrent	8	iid	105	−3.81	[−13.33, +5.71]	−9 / +2 / +3	13 / 17	0.58	no
prenorm-loop	latent-recurrent	8	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
prenorm-loop	latent-recurrent	8	ood_operand	96	−3.12	[−10.42, +4.17]	−4 / +2 / −1	5 / 8	0.58	no
prenorm-loop	latent-recurrent	16	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
prenorm-loop	latent-recurrent	16	ood_operand	96	−6.25	[−12.50, +0.00]	−4 / +0 / −2	2 / 8	0.11	no
arloop-untied	latent-recurrent	1	iid	105	−3.81	[−12.38, +4.76]	−7 / +1 / +2	9 / 13	0.52	no
arloop-untied	latent-recurrent	1	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-untied	latent-recurrent	1	ood_operand	96	−6.25	[−13.54, +1.04]	−3 / −2 / −1	4 / 10	0.18	no
arloop-untied	latent-recurrent	2	iid	105	−5.71	[−14.29, +2.86]	−5 / −1 / +0	7 / 13	0.26	no
arloop-untied	latent-recurrent	2	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no

X	Y	R	SPLIT	N	X - Y	95% INTERVAL	PER SEED	DISCORDANT	MCNEMAR P	CANDIDATE
arloop-untied	latent-recurrent	2	ood_operand	96	−6.25	[−13.54, +1.04]	−3 / −1 / −2	4 / 10	0.18	no
arloop-untied	latent-recurrent	4	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-untied	latent-recurrent	4	ood_operand	96	−5.21	[−12.50, +2.08]	−3 / −1 / −1	4 / 9	0.27	no
arloop-untied	latent-recurrent	8	iid	105	−10.48	[−19.05, −1.90]	−10 / −1 / +0	7 / 18	0.043	no
arloop-untied	latent-recurrent	8	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-untied	latent-recurrent	8	ood_operand	96	−6.25	[−13.54, +0.00]	−4 / −1 / −1	3 / 9	0.15	no
arloop-untied	latent-recurrent	16	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-untied	latent-recurrent	16	ood_operand	96	−6.25	[−13.54, +0.00]	−3 / −2 / −1	3 / 9	0.15	no
arloop-tied	latent-recurrent	1	iid	105	−3.81	[−12.38, +4.76]	−11 / +2 / +5	10 / 14	0.54	no
arloop-tied	latent-recurrent	1	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-tied	latent-recurrent	1	ood_operand	96	−5.21	[−12.50, +1.04]	−3 / +0 / −2	3 / 8	0.23	no
arloop-tied	latent-recurrent	2	iid	105	−4.76	[−13.33, +3.81]	−10 / +2 / +3	10 / 15	0.42	no
arloop-tied	latent-recurrent	2	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-tied	latent-recurrent	2	ood_operand	96	−7.29	[−13.54, −1.04]	−3 / +0 / −4	2 / 9	0.065	no
arloop-tied	latent-recurrent	4	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-tied	latent-recurrent	4	ood_operand	96	−7.29	[−13.54, −1.04]	−3 / +0 / −4	2 / 9	0.065	no
arloop-tied	latent-recurrent	8	iid	105	−4.76	[−13.33, +3.81]	−10 / +1 / +4	10 / 15	0.42	no
arloop-tied	latent-recurrent	8	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-tied	latent-recurrent	8	ood_operand	96	−6.25	[−12.50, +0.00]	−3 / +0 / −3	2 / 8	0.11	no
arloop-tied	latent-recurrent	16	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-tied	latent-recurrent	16	ood_operand	96	−6.25	[−12.50, +0.00]	−3 / +0 / −3	2 / 8	0.11	no

SECONDARY: AGAINST THE DEPTH-MATCHED CONTROL (63)

latent-recurrent	standard-depth	1	iid	105	+2.86	[−4.76, +11.43]	+8 / −3 / −2	12 / 9	0.66	no
latent-recurrent	standard-depth	1	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
latent-recurrent	standard-depth	1	ood_operand	96	+9.38	[+3.12, +15.62]	+4 / +2 / +3	10 / 1	0.012	**yes**
latent-recurrent	standard-depth	2	iid	105	+3.81	[−3.81, +11.43]	+7 / −3 / +0	12 / 8	0.5	no
latent-recurrent	standard-depth	2	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
latent-recurrent	standard-depth	2	ood_operand	96	+9.38	[+3.12, +15.62]	+4 / +1 / +4	10 / 1	0.012	**yes**
latent-recurrent	standard-depth	4	iid	105	+5.71	[−1.90, +13.33]	+7 / −1 / +0	13 / 7	0.26	no
latent-recurrent	standard-depth	4	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
latent-recurrent	standard-depth	4	ood_operand	96	+9.38	[+3.12, +15.62]	+4 / +1 / +4	10 / 1	0.012	**yes**
latent-recurrent	standard-depth	8	iid	105	+4.76	[−3.81, +13.33]	+7 / −2 / +0	13 / 8	0.38	no
latent-recurrent	standard-depth	8	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
latent-recurrent	standard-depth	8	ood_operand	96	+9.38	[+3.12, +15.62]	+4 / +1 / +4	10 / 1	0.012	**yes**
latent-recurrent	standard-depth	16	iid	105	+4.76	[−3.81, +13.33]	+7 / −2 / +0	13 / 8	0.38	no
latent-recurrent	standard-depth	16	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
latent-recurrent	standard-depth	16	ood_operand	96	+9.38	[+3.12, +15.62]	+4 / +1 / +4	10 / 1	0.012	**yes**

X	Y	R	SPLIT	N	X - Y	95% INTERVAL	PER SEED	DISCORDANT	MCNEMAR P	CANDIDATE
prenorm-loop	standard-depth	1	iid	105	−3.81	[−11.43, +3.81]	−1 / −5 / +2	7 / 11	0.48	no
prenorm-loop	standard-depth	1	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
prenorm-loop	standard-depth	1	ood_operand	96	+8.33	[+3.12, +13.54]	+0 / +3 / +5	8 / 0	0.0078	no
prenorm-loop	standard-depth	2	iid	105	−0.95	[−9.52, +7.62]	−1 / −3 / +3	10 / 11	1	no
prenorm-loop	standard-depth	2	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
prenorm-loop	standard-depth	2	ood_operand	96	+7.29	[+3.12, +12.50]	+0 / +3 / +4	7 / 0	0.016	no
prenorm-loop	standard-depth	4	iid	105	+0.00	[−7.62, +7.62]	−1 / −2 / +3	9 / 9	1	no
prenorm-loop	standard-depth	4	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
prenorm-loop	standard-depth	4	ood_operand	96	+6.25	[+2.08, +11.46]	+0 / +3 / +3	6 / 0	0.03125	no
prenorm-loop	standard-depth	8	iid	105	+0.95	[−7.62, +9.52]	−2 / +0 / +3	11 / 10	1	no
prenorm-loop	standard-depth	8	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
prenorm-loop	standard-depth	8	ood_operand	96	+6.25	[+2.08, +11.46]	+0 / +3 / +3	6 / 0	0.03125	no
prenorm-loop	standard-depth	16	iid	105	−1.90	[−10.48, +5.71]	−2 / +0 / +0	9 / 11	0.82	no
prenorm-loop	standard-depth	16	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
prenorm-loop	standard-depth	16	ood_operand	96	+3.12	[−1.04, +8.33]	+0 / +1 / +2	4 / 1	0.375	no
arloop-untied	standard-depth	1	iid	105	−0.95	[−8.57, +6.67]	+1 / −2 / +0	9 / 10	1	no
arloop-untied	standard-depth	1	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-untied	standard-depth	1	ood_operand	96	+3.12	[−1.04, +8.33]	+1 / +0 / +2	4 / 1	0.375	no
arloop-untied	standard-depth	2	iid	105	−1.90	[−9.52, +5.71]	+2 / −4 / +0	8 / 10	0.81	no
arloop-untied	standard-depth	2	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-untied	standard-depth	2	ood_operand	96	+3.12	[−1.04, +8.33]	+1 / +0 / +2	4 / 1	0.375	no
arloop-untied	standard-depth	4	iid	105	−0.95	[−9.52, +6.67]	+1 / −3 / +1	9 / 10	1	no
arloop-untied	standard-depth	4	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-untied	standard-depth	4	ood_operand	96	+4.17	[+0.00, +9.38]	+1 / +0 / +3	5 / 1	0.22	no
arloop-untied	standard-depth	8	iid	105	−5.71	[−14.29, +2.86]	−3 / −3 / +0	7 / 13	0.26	no
arloop-untied	standard-depth	8	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-untied	standard-depth	8	ood_operand	96	+3.12	[−1.04, +7.29]	+0 / +0 / +3	4 / 1	0.375	no
arloop-untied	standard-depth	16	iid	105	−2.86	[−11.43, +5.71]	+1 / −3 / −1	9 / 12	0.66	no
arloop-untied	standard-depth	16	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-untied	standard-depth	16	ood_operand	96	+3.12	[−1.04, +7.29]	+1 / −1 / +3	4 / 1	0.375	no
arloop-tied	standard-depth	1	iid	105	−0.95	[−10.48, +7.62]	−3 / −1 / +3	11 / 12	1	no
arloop-tied	standard-depth	1	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-tied	standard-depth	1	ood_operand	96	+4.17	[+1.04, +8.33]	+1 / +2 / +1	4 / 0	0.125	**yes**

X	Y	R	SPLIT	N	X - Y	95% INTERVAL	PER SEED	DISCORDANT	MCNEMAR P	CANDIDATE
arloop-tied	standard-depth	2	iid	105	−0.95	[−9.52, +7.62]	−3 / −1 / +3	10 / 11	1	no
arloop-tied	standard-depth	2	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-tied	standard-depth	2	ood_operand	96	+2.08	[+0.00, +5.21]	+1 / +1 / +0	2 / 0	0.5	no
arloop-tied	standard-depth	4	iid	105	−0.95	[−9.52, +7.62]	−3 / −1 / +3	10 / 11	1	no
arloop-tied	standard-depth	4	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-tied	standard-depth	4	ood_operand	96	+2.08	[+0.00, +5.21]	+1 / +1 / +0	2 / 0	0.5	no
arloop-tied	standard-depth	8	iid	105	+0.00	[−8.57, +8.57]	−3 / −1 / +4	11 / 11	1	no
arloop-tied	standard-depth	8	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-tied	standard-depth	8	ood_operand	96	+3.12	[+0.00, +7.29]	+1 / +1 / +1	3 / 0	0.25	no
arloop-tied	standard-depth	16	iid	105	+0.00	[−8.57, +8.57]	−3 / −1 / +4	11 / 11	1	no
arloop-tied	standard-depth	16	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-tied	standard-depth	16	ood_operand	96	+3.12	[+0.00, +7.29]	+1 / +1 / +1	3 / 0	0.25	no
standard-parameter	standard-depth	–	iid	105	−14.29	[−21.90, −7.62]	−4 / −7 / −4	1 / 16	2.7×10^{-4}	**yes**
standard-parameter	standard-depth	–	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
standard-parameter	standard-depth	–	ood_operand	96	−1.04	[−3.12, +0.00]	+0 / −1 / +0	0 / 1	1	no
SECONDARY: THE DEEPEST AGAINST THE TRAINING MEAN, WITHIN AN ARM (12)										
latent-recurrent	latent-recurrent	16 vs 4	iid	105	−0.95	[−2.86, +0.00]	+0 / −1 / +0	0 / 1	1	no
latent-recurrent	latent-recurrent	16 vs 4	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
latent-recurrent	latent-recurrent	16 vs 4	ood_operand	96	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
prenorm-loop	prenorm-loop	16 vs 4	iid	105	−1.90	[−7.62, +2.86]	−1 / +2 / −3	3 / 5	0.73	no
prenorm-loop	prenorm-loop	16 vs 4	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
prenorm-loop	prenorm-loop	16 vs 4	ood_operand	96	−3.12	[−7.29, +0.00]	+0 / −2 / −1	0 / 3	0.25	no
arloop-untied	arloop-untied	16 vs 4	iid	105	−1.90	[−5.71, +1.90]	+0 / +0 / −2	1 / 3	0.625	no
arloop-untied	arloop-untied	16 vs 4	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-untied	arloop-untied	16 vs 4	ood_operand	96	−1.04	[−5.21, +2.08]	+0 / −1 / +0	1 / 2	1	no
arloop-tied	arloop-tied	16 vs 4	iid	105	+0.95	[+0.00, +2.86]	+0 / +0 / +1	1 / 0	1	no
arloop-tied	arloop-tied	16 vs 4	ood_depth	192	+0.00	[+0.00, +0.00]	+0 / +0 / +0	0 / 0	1	no
arloop-tied	arloop-tied	16 vs 4	ood_operand	96	+1.04	[+0.00, +3.12]	+0 / +0 / +1	1 / 0	1	no

A.5 E3: exploratory paired diagnostics

TABLE 26 E3, post hoc: accepted actions per attempt (paired sign test over targets) and single-step validity (McNemar over rows), from the re-check.

ARM AND LOOPS	ARLOOP AGAINST PRENORM
actions:native3, R = 1	mean +0.00 actions; ARLoop longer on 0 targets, shorter on 0; sign test p 1
actions:native3, R = 2	mean +0.00 actions; ARLoop longer on 0 targets, shorter on 0; sign test p 1
actions:native3, R = 3	mean +1.19 actions; ARLoop longer on 11 targets, shorter on 10; sign test p 1
actions:native3, R = 4	mean +5.47 actions; ARLoop longer on 27 targets, shorter on 5; sign test p 1.1×10^{-4}
actions:random124, R = 1	mean +5.72 actions; ARLoop longer on 31 targets, shorter on 0; sign test p 9.3×10^{-10}
actions:random124, R = 2	mean +9.53 actions; ARLoop longer on 26 targets, shorter on 1; sign test p 4.2×10^{-7}
actions:random124, R = 3	mean −3.38 actions; ARLoop longer on 9 targets, shorter on 14; sign test p 0.4
actions:random124, R = 4	mean +12.84 actions; ARLoop longer on 31 targets, shorter on 0; sign test p 9.3×10^{-10}
steps:native3, R = 1	valid 0 against 0; only ARLoop 0, only PreNorm 0; McNemar p 1
steps:native3, R = 2	valid 0 against 0; only ARLoop 0, only PreNorm 0; McNemar p 1

ARM AND LOOPS	ARLOOP AGAINST PRENORM
steps:native3, R = 3	valid 49 against 44; only ARLoop 5, only PreNorm 0; McNemar p 0.0625
steps:native3, R = 4	valid 47 against 40; only ARLoop 10, only PreNorm 3; McNemar p 0.092
steps:random124, R = 1	valid 43 against 36; only ARLoop 11, only PreNorm 4; McNemar p 0.12
steps:random124, R = 2	valid 45 against 39; only ARLoop 6, only PreNorm 0; McNemar p 0.03125
steps:random124, R = 3	valid 45 against 40; only ARLoop 6, only PreNorm 1; McNemar p 0.125
steps:random124, R = 4	valid 45 against 39; only ARLoop 6, only PreNorm 0; McNemar p 0.03125

A.6 Stage 1a: the full ladder

TABLE 27 Stage 1a, every suite: accepted proofs (of 32) · accepted actions per attempt · valid single steps (of 64) · dev cross-entropy (nats per token). The 512-update rung is E3.

ADAPTER	UPDATES	R = 1	R = 2	R = 3	R = 4
ARLoop-0.6B, native3	512	0 · 0.0 · 0 · 11.653	0 · 0.0 · 0 · 8.572	0 · 16.5 · 49 · 0.379	0 · 12.2 · 47 · 0.570
ARLoop-0.6B, native3	1,024	0 · 0.0 · 0 · 11.945	0 · 0.0 · 0 · 8.488	6 · 23.7 · 62 · 0.229	0 · 19.4 · 58 · 0.269
ARLoop-0.6B, native3	2,048	0 · 0.0 · 0 · 12.267	0 · 0.0 · 0 · 8.928	2 · 24.8 · 62 · 0.226	0 · 23.5 · 59 · 0.252
ARLoop-0.6B, native3	4,096	0 · 0.0 · 0 · 12.527	0 · 0.0 · 0 · 9.764	4 · 27.0 · 63 · 0.324	0 · 25.5 · 59 · 0.351
ARLoop-0.6B, random124	512	0 · 6.7 · 43 · 0.486	0 · 13.0 · 45 · 0.474	0 · 18.4 · 45 · 0.446	0 · 17.0 · 45 · 0.445
ARLoop-0.6B, random124	1,024	0 · 10.2 · 49 · 0.410	0 · 13.0 · 50 · 0.402	0 · 11.1 · 49 · 0.372	0 · 9.4 · 49 · 0.373
ARLoop-0.6B, random124	2,048	0 · 19.7 · 54 · 0.320	0 · 20.9 · 56 · 0.273	0 · 20.1 · 58 · 0.254	0 · 20.9 · 60 · 0.253
ARLoop-0.6B, random124	4,096	0 · 21.5 · 55 · 0.281	2 · 23.9 · 61 · 0.242	3 · 24.7 · 61 · 0.235	3 · 25.7 · 61 · 0.233
PreNormLoop-0.6B, native3	512	0 · 0.0 · 0 · 8.982	0 · 0.0 · 0 · 6.191	0 · 15.3 · 44 · 0.395	0 · 6.8 · 40 · 0.834
PreNormLoop-0.6B, native3	1,024	0 · 0.0 · 0 · 9.318	0 · 0.0 · 0 · 6.481	1 · 18.3 · 61 · 0.246	0 · 4.8 · 52 · 0.557
PreNormLoop-0.6B, native3	2,048	0 · 0.0 · 0 · 9.340	0 · 0.0 · 0 · 6.566	5 · 27.4 · 63 · 0.212	1 · 7.2 · 53 · 0.593
PreNormLoop-0.6B, native3	4,096	0 · 0.0 · 0 · 9.285	0 · 0.0 · 0 · 6.505	3 · 27.3 · 62 · 0.325	0 · 8.2 · 51 · 0.621
PreNormLoop-0.6B, random124	512	0 · 1.0 · 36 · 0.520	0 · 3.5 · 39 · 0.457	0 · 21.8 · 40 · 0.459	0 · 4.2 · 39 · 0.458
PreNormLoop-0.6B, random124	1,024	0 · 10.3 · 49 · 0.392	0 · 14.0 · 50 · 0.360	0 · 18.5 · 48 · 0.359	0 · 10.7 · 51 · 0.363
PreNormLoop-0.6B, random124	2,048	0 · 18.2 · 51 · 0.350	0 · 21.3 · 53 · 0.328	0 · 21.4 · 52 · 0.336	0 · 21.2 · 51 · 0.336
PreNormLoop-0.6B, random124	4,096	0 · 22.1 · 49 · 0.334	0 · 21.6 · 50 · 0.330	0 · 22.0 · 52 · 0.342	0 · 22.1 · 51 · 0.345

A.7 Stage 1c: rules 1 and 2

TABLE 28 Stage 1c, the two registered pairs and every arm against E1's released model.

COMPARISON	A	B	A - B	95% INTERVAL	MCNEMAR P
L4 vs C3	37.91%	38.51%	−0.61	[−2.20, +1.06]	0.52
L4e vs C3e	38.13%	38.51%	−0.38	[−2.05, +1.36]	0.72
L4 vs ar06-R3	37.91%	38.97%	−1.06	[−2.50, +0.45]	0.19
L4 vs ar06-R4-cycle	37.91%	38.82%	−0.91	[−2.58, +0.76]	0.33
C3 vs ar06-R3	38.51%	38.97%	−0.45	[−1.59, +0.68]	0.52
L4e vs ar06-R3	38.13%	38.97%	−0.83	[−2.35, +0.76]	0.33
L4e vs ar06-R4-cycle	38.13%	38.82%	−0.68	[−2.43, +0.99]	0.49
C3e vs ar06-R3	38.51%	38.97%	−0.45	[−1.82, +0.91]	0.59

B APPENDIX

The protocols in brief

Each protocol's sections, as registered and amended (each amendment is a section of its own), with its first commit. The decision rules are quoted in full in the sections of the report. The confirmatory proposal's "Stage 2: confirmatory" is this report's Stage 4.

E1 · experiments/e1_release/PROTOCOL.md

First commit 379b98a 2026-11-15 03:30 CET. Sections: Models (pinned); Evaluation (the paper's protocol, App. B / lm_eval_tasks/arloop_largescale.yaml); Loop counts; Questions and decision rules (paired over the same 1,319 items); What is not done; Amendments (before any full run); Additions after full runs (exploratory, no decision rule).

E2 · experiments/e2_small/PROTOCOL.md

First commit 4fcdd39 2026-11-15 03:35 CET. Sections: 1. Question; 2. Sources (pinned); 3. Task, data, training budget (identical to the pilot for every arm); 4. Arms; 5. Evaluation; 6. Comparisons and what counts as an effect; 7. Checks before the main run (smoke job); 8. Reproduction of job 284986 by the unchanged arm; 9. Compute and bookkeeping.

E3 · experiments/e3_pair/PROTOCOL.md

First commit 4fcdd39 2026-11-15 03:35 CET. Sections: Question; Bases (frozen, loaded through ../e1_release/loopctl.py, BF16); Arms (2 bases x 2 arms = 4 fine-tuned checkpoints); Data; Training recipe (C4 main, configs/campaign4-main.json; identical for all four runs); Evaluation (eval_proofs.py); Gates (all must pass before the first training job); Comparisons (fixed now; all descriptive at n = 32, one seed); Engineering success; Compute and rules.

STAGE 1A · experiments/e3_pair/PROTOCOL-ladder.md

First commit dbef249 2026-11-15 12:56 CET. Sections: Question; Design (as in E3, PROTOCOL.md, unless listed here); Decision rules (fixed now); Records; Amendment 1 (2026-11-15 13:23 CET, before any Stage 1a result exists; the trainings had just started).

STAGE 1B · experiments/e2_small/PROTOCOL-long.md

First commit dbef249 2026-11-15 12:56 CET. Sections: Question; Design; Decision rules (fixed now), applied to each length separately; Compute and records.

STAGE 1C · experiments/e1_release/PROTOCOL-loop4.md

First commit 1a36937 2026-11-15 13:00 CET. Sections: Question; Design; Decision rules (fixed now); Compute and records; Amendment 1 (2026-11-15 13:56 CET, before any Stage 1c result; the registered trainings L4 and C3 had run about an hour).

E3 CONFIRMATORY PROPOSAL · experiments/e3_pair/CONFIRM-PROPOSAL.md

First commit 86acfbc 2026-11-15 08:48 CET. Sections: Why; Stage 1: get off the floor (one seed, dev only); Stage 2: confirmatory (fixed before Stage 2 starts); Gates and records (as in the discovery run).

THE CAMPAIGN PLAN · CAMPAIGN.md

First commit 0263d51 2026-11-15 05:17 CET. Sections: The question; Stage 1 — now, WMI (one day; results could reach the talk only if started today); Stage 2 — after the talk, Helios GH200: continued training at depth (k.depth-training); Stage 3 — after the talk, Helios GH200: from scratch at 140M, a token ladder (k.token-gain); Stage 4 — confirmatory proofs (only if 1a leaves the floor); LUMI (MI250X); Cost basis (computed 2026-11-15 from the job records); Gates before any stage.

STAGE 4 · experiments/e3_pair/PROTOCOL-confirm.md

First commit 720cab6 2026-11-16 00:06 CET. Sections: Questions; Design (as in Stage 1a unless listed here); Data: the sealed C4 test split, opened once; Evaluation (E3's harness, unchanged: C4 budgets, greedy KV-cached decoding, 600 s per target, the pinned Python checker; not Lean); Decision rules (fixed now); Compute (estimates from Stage 1a's job records); Records; Amendment 1 (2026-11-16 00:15 CET, before any Stage 4 training or evaluation; nothing was measured).

STAGE 2 · experiments/s2_depth/PROTOCOL.md

First commit 86c6f82 2026-11-16 00:05 CET. Sections: Question; Deviations from CAMPAIGN.md Stage 2 (decided before any run, with the reasons); Design; Gates (before any training run counts); Decision rules (fixed now); Records; Amendment 1 (2026-11-16 00:17 CET, before any main run; only gates and speed measurements have run); Amendment 2 (2026-11-16 11:41 CET, before the random-depth arm runs; no Stage 2 result has been read).

C APPENDIX

The research map's test cells

Each planned test cell of E1, E2 and E3, with the status set after the re-checks. The map's own definitions: unknown (not measured); supports (measured, supports the claim or ARLoop is better); contradicts (measured, contradicts); inconclusive (measured, interval includes 0). The one cell marked "contradicts" (e2.tied-depth, a gain of tied queries from R = 16 over R = 4) has identical verdicts at the two depths on every item, so the measured gain is exactly 0, which the map reads as contradicting a gain. Stage 1 and the later stages are evidence nodes of the map, not cells of this grid.

TABLE 29 Every planned test cell (labyrinth/cells.json).

CELL	TEST	STATUS
E1 RELEASED CHECKPOINTS		
e1.pair-gsm8k	ARLoop-0.6B > PreNormLoop-0.6B on GSM8K (paired)	SUPPORTS
e1.pair-nobos	Control: ARLoop-0.6B > PreNormLoop-0.6B on GSM8K without BOS (paired)	INCONCLUSIVE
e1.1p2b-gsm8k	ARLoop-1.2B GSM8K within 2 pts of 63.1	SUPPORTS
e1.ar-r4-repeat	ARLoop-0.6B: no gain at R = 4 (repeat rule) over R = 3	SUPPORTS
e1.ar-r4-cycle	ARLoop-0.6B: no gain at R = 4 (cycle rule) over R = 3	SUPPORTS
e1.pn-r4	PreNormLoop-0.6B: no gain at R = 4 over R = 3	SUPPORTS
e1.pn-r5	PreNormLoop-0.6B: no gain at R = 5 over R = 3	SUPPORTS
e1.pn-r6	PreNormLoop-0.6B: no gain at R = 6 over R = 3	SUPPORTS
e1.ar-r2	ARLoop-0.6B: R = 3 better than R = 2	SUPPORTS

CELL	TEST	STATUS
e1.pn-r2	PreNormLoop-0.6B: R = 3 better than R = 2	SUPPORTS
e1.ar-r5-repeat	ARLoop-0.6B: no gain at R = 5 (repeat rule) over R = 3	SUPPORTS
e1.ar-r6-repeat	ARLoop-0.6B: no gain at R = 6 (repeat rule) over R = 3	SUPPORTS
e1.ar12-r4-repeat	ARLoop-1.2B: no gain at R = 4 (repeat rule) over R = 3	SUPPORTS
E2 CANARY (LEAN-CHECKED)		
e2.prenorm-iid	prenorm-loop vs latent-recurrent, iid (R = 4 and 16)	INCONCLUSIVE
e2.arloop-untied-iid	arloop-untied vs latent-recurrent, iid (R = 4 and 16)	INCONCLUSIVE
e2.arloop-tied-iid	arloop-tied vs latent-recurrent, iid (R = 4 and 16)	INCONCLUSIVE
e2.arloop-untied-ood	arloop-untied vs latent-recurrent, ood_operand (R = 4 and 16)	INCONCLUSIVE
e2.arloop-tied-ood	arloop-tied vs latent-recurrent, ood_operand (R = 4 and 16)	INCONCLUSIVE
e2.tied-depth	arloop-tied: gain from R = 16 over R = 4	CONTRADICTS
E3 CHECKED PROOFS (CAMPAIGN 4 DEV)		
e3.native3-L1	native3 arm, R = 1: ARLoop vs PreNormLoop	INCONCLUSIVE
e3.native3-L2	native3 arm, R = 2	INCONCLUSIVE
e3.native3-L3	native3 arm, R = 3	INCONCLUSIVE
e3.native3-L4	native3 arm, R = 4	INCONCLUSIVE
e3.random-L1	random124 arm, R = 1	INCONCLUSIVE
e3.random-L2	random124 arm, R = 2	INCONCLUSIVE
e3.random-L3	random124 arm, R = 3	INCONCLUSIVE
e3.random-L4	random124 arm, R = 4	INCONCLUSIVE

D APPENDIX

Glossary

loop one pass through the shared stack of layers.

R the number of loops; the experiments' code calls it r (E2) or L (E3, Stage 1a).

effective layer a layer of the unrolled model; a model has as many as its physical layers times its loops.

sublayer the attention or the MLP part of an effective layer; ARLoop routes per sublayer.

residual stream the running state that every sublayer reads from and writes to.

routing ARLoop's attention over depth: a softmax over the embedding and the residual blocks.

routing query the learned vector of a sublayer that scores the sources; one per effective sublayer.

read-out the last routing call, before the output head.

block consecutive sublayers whose updates are summed, as in PreNorm, before routing sees them.

window the sources kept for routing: the embedding and the 3 most recent blocks.

the matched pair ARLoop-0.6B and PreNormLoop-0.6B: same data, tokeniser, backbone and budget.

the canary E2's toy task: closed addition with a Lean-checked certificate.

checked proof a proof accepted by the pinned constructive Python checker of `latent` (E3, Stage 1a, Stage 4); its acceptance is a kernel-checked QED (the end of a proof the checker's kernel verifies). E2's outputs are certificates, checked by Lean.

row	one item of `latent`'s proof corpus: a proof state with its gold next action. Rows come from proof sessions and are split into train, development and test.
roster	a frozen list of rows on which single-step validity is measured.
single-step validity	one greedy action from the gold state of a row; valid if the checker accepts the transition. Measured on a roster.
iid, ood_depth, ood_operand	E2's evaluation splits: in-distribution, out-of-distribution depth (longer carries), out-of-distribution operands.
ar06-R3, pn06-R4, ar12-R2	E1's run names: model (ARLoop-0.6B, PreNormLoop-0.6B, ARLoop-1.2B) and number of loops; a suffix names the rule for missing routing parameters (repeat, cycle).
k…., h…., q….	identifiers in the research map: conjectures (k), hunches (h), open questions (q).
b_1 … b_6	ARLoop-0.6B's residual blocks at three loops, two per loop; the sixth stays the open (partial) block.
CE	cross-entropy, in nats per token.
Rrand	Stage 2's random-depth arm (R drawn per update; PreNorm only, by amendment 2).
anytime model	a model usable at any number of loops.
the proof floor	no proof (or almost none) accepted in any suite, so no comparison can be decided.
paired comparison	both sides scored on the same items (and seeds, data and batches).
E1's paired statistics	the paired bootstrap interval and the exact McNemar test of section 3.1, as later protocols cite them.
re-check	a recount from the raw outputs with code written apart from the analysis (section 3.3 lists the exceptions).
BOS	the beginning-of-sequence token, prepended by the authors' evaluation script.
Holm	a step-down correction of the p-values of a family of tests.
candidate effect	E2's discovery rule: the interval excludes 0 and every seed agrees in sign.
random124	E3's random-depth arm: each update draws its number of loops from {1, 2, 4}.
native3	E3's arm trained at the released models' three loops.

E APPENDIX

References

1. Yuekun Yao, Yupei Du, Dawei Zhu, Huan Sun. *ARLoop: Fully-Open Pre-trained Looped Transformers with Attention Residuals*, arXiv:2611.05234 (v1, 3 November 2026), 2026.
2. Kimi. *Attention Residuals*, By the Kimi Team, arXiv:2603.15031, 2026.
3. Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer et al. *Scaling up Test-Time Compute with Latent Reasoning: A Recurrent Depth Approach*, arXiv:2502.05171, 2025.
4. Rui-Jie Zhu, Zixuan Wang, Kai Hua, Tianyu Zhang et al. *Scaling Latent Reasoning via Looped Language Models*, arXiv:2510.25741, 2025.
5. Hayden Prairie, Zachary Novack, Taylor Berg-Kirkpatrick, Daniel Y. Fu. *Parcae: Scaling Laws For Stable Looped Language Models*, arXiv:2604.12946, 2026.
6. Sean McLeish, Ang Li, John Kirchenbauer, Dayal Singh Kalra et al. *Teaching Pretrained Language Models to Think Deeper with Retrofitted Recurrence*, arXiv:2511.07384, 2025.
7. Tsung-Han Wu, Minsuk Heo, Kenny Tan. *Do Halting Heads Really Save Compute in Looped Language Models?*, arXiv:2610.32246, 2026.

8. Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit et al. *Universal Transformers*, International Conference on Learning Representations, 2019.
9. Liu Yang, Kangwook Lee, Robert Nowak, Dimitris Papailiopoulos. *Looped Transformers are Better at Learning Learning Algorithms*, International Conference on Learning Representations, 2024.
10. Nikunj Saunshi, Nishanth Dikkala, Zhiyuan Li, Sanjiv Kumar et al. *Reasoning with Latent Thoughts: On the Power of Looped Transformers*, International Conference on Learning Representations, 2025.
11. Shuzhen Li, Yifan Zhang, Jiacheng Guo, Quanquan Gu et al. *DeepLoop: Depth Scaling for Looped Transformers*, arXiv:2607.13491, 2026.
12. Abbas Zeitoun, Lucas Torroba-Hennigen, Yoon Kim. *Hyperloop Transformers*, arXiv:2604.21254, 2026.
13. Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng et al. *On Layer Normalization in the Transformer Architecture*, arXiv:2002.04745, 2020.
14. Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang et al. *DeepNet: Scaling Transformers to 1,000 Layers*, arXiv:2203.00555, 2022.
15. Arpit Bansal, Avi Schwarzschild, Eitan Borgnia, Zeyad Emam et al. *End-to-end Algorithm Synthesis with Recurrent Networks: Logical Extrapolation Without Overthinking*, arXiv:2202.05826 (NeurIPS 2022), 2022.
16. Zhanchao Zhou, Tianyi Wu, Zhiyun Jiang, Fares Obeid et al. *Value Residual Learning*, arXiv:2410.17897, first posted October 2024, 2025.
17. Róbert Csordás, Christopher D. Manning, Christopher Potts. *Do Language Models Use Their Depth Efficiently?*, arXiv:2505.13898, 2025.
18. Harsh Kohli, Srinivasan Parthasarathy, Huan Sun, Yuekun Yao. *Loop, Think, & Generalize: Implicit Reasoning in Recurrent-Depth Transformers*, arXiv:2604.07822, 2026.
19. Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen et al. *Training Verifiers to Solve Math Word Problems*, arXiv:2110.14168, 2021.
20. Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman et al. *The Language Model Evaluation Harness*, Zenodo, doi:10.5281/zenodo.12608602, 2024.

COLOPHON

Sources of this report

Built 2026-11-17 01:31 CET by `report/build_report.py` at commit `12bbf1d`. Typography and colours: `DESIGN.md` (Instrument Serif, Inter, JetBrains Mono, JuliaMono; formulas by MathJax). The files read, each with the start of its SHA-256:

FILE	SHA-256
CAMPAIGN.md	8dd01de8823b3d82…
DESIGN.md	0eedcbe9801b4cd8…
artifacts/mathematical/digest.md	f3d9b246c5c222e4…
book/references.bib	4acce90966bd9481…
experiments/e1_release/PROTOCOL-loop4.md	54a19b534a9fe9d2…
experiments/e1_release/PROTOCOL.md	496d12ff237b38be…
experiments/e1_release/analyze.py	a756b03ddff3b0bf…
experiments/e1_release/analyze_loop4.py	cda9c615d49a54bb…
experiments/e1_release/full-A.cmds	8886571f697e170a…
experiments/e2_small/PROTOCOL-long.md	f9f5d1b10a1f1990…
experiments/e2_small/PROTOCOL.md	5f4b04bae42994bf…
experiments/e2_small/results/322153/analysis.json	4de9b809388562c9…
experiments/e2_small/results/322153/summary.json	4fe47fd9f0ee8450…

FILE	SHA-256
experiments/e2_small/results/322560/analysis.json	579c1911eed5cc1c…
experiments/e2_small/results/322560/summary.json	d38fe00047427d82…
experiments/e2_small/results/322667/analysis.json	4fddeda99bd0f061…
experiments/e2_small/results/322667/summary.json	ccd977cfd2c16d8f…
experiments/e2_small/results/jobs_sacct.txt	fd05e480f63762a9…
experiments/e2_small/scripts/analyze.py	108474ef1aac5dd3…
experiments/e3_pair/CONFIRM-PROPOSAL.md	a5ff8619e1ffd5cf…
experiments/e3_pair/PROTOCOL-confirm.md	80e56dcc22a992c5…
experiments/e3_pair/PROTOCOL-ladder.md	a3c44e20245a5ff3…
experiments/e3_pair/PROTOCOL.md	bc6051df270fa4a6…
experiments/e3_pair/analyze.py	d6f9378846cbf77b…
experiments/e3_pair/analyze_ladder.py	826acb486f5ee5d5…
experiments/e3_pair/results/data/c4-smollm2-v1/audit.json	1b4570912603c576…
experiments/s2_depth/PROTOCOL.md	d3753dc836b1d9d4…
labyrinth/cells.json	8ed89e625e8edfd0…
labyrinth/lab.py	91d1d5d9ba403379…
runs/e1/bos-check/ar06-R3-bos/receipt.json	ba23386e6a4dc4a0…
runs/e1/bos-check/ar06-R3-nobos/receipt.json	0067e57e375d15da…
runs/e1/bos-check/pn06-R3-bos/receipt.json	8f8f3c29028886c5…
runs/e1/bos-check/pn06-R3-nobos/receipt.json	84faa91e31d26add…
runs/e1/gsm8k-bos-repeat/pn06-R3/receipt.json	6e89bd8d4f30b70a…
runs/e1/gsm8k-bos/ar06-R1/receipt.json	02fb98fd5c73abda…
runs/e1/gsm8k-bos/ar06-R2/receipt.json	2c01123021f6746a…
runs/e1/gsm8k-bos/ar06-R3/receipt.json	5852082d6312314f…
runs/e1/gsm8k-bos/ar06-R4-cycle/receipt.json	dbaabfc4f8d13c6c…
runs/e1/gsm8k-bos/ar06-R4-repeat/receipt.json	f7ec4af3c3a71f87…
runs/e1/gsm8k-bos/ar06-R5-repeat/receipt.json	9cd907bca9cc5c87…
runs/e1/gsm8k-bos/ar06-R6-repeat/receipt.json	921fdc4a9f883df0…
runs/e1/gsm8k-bos/ar12-R2/receipt.json	1e12b5de551516c9…
runs/e1/gsm8k-bos/ar12-R3/receipt.json	440e3635e91fbf17…
runs/e1/gsm8k-bos/ar12-R4-repeat/receipt.json	88dc2dcf6d7b88ad…
runs/e1/gsm8k-bos/pn06-R1/receipt.json	af355767f027c9b0…
runs/e1/gsm8k-bos/pn06-R2/receipt.json	d3e45f32c9dc3470…
runs/e1/gsm8k-bos/pn06-R3/receipt.json	45fb60927aaa8b81…
runs/e1/gsm8k-bos/pn06-R4/receipt.json	086f9029c74993cc…
runs/e1/gsm8k-bos/pn06-R5/receipt.json	51d0ba8ab9dba5e5…
runs/e1/gsm8k-bos/pn06-R6/receipt.json	5ddeb3b46c8bfe98…
runs/e1/gsm8k-nobos-rerun/ar06-R3/receipt.json	12e3db57ae1f0850…
runs/e1/gsm8k-nobos/ar06-R3/receipt.json	b13832c5a106859d…
runs/e1/gsm8k-nobos/pn06-R3/receipt.json	bc41096952df5267…
runs/e1/referee.md	5d37a36ca585473d…
runs/e1/referee/e1_referee.json	90ab7bd922ada2ef…
runs/e1/referee/e1_referee.py	7e28f39785fd40c2…
runs/e1/routing/ar06-R3/routing.json	098a6735e029e61d…
runs/e1/routing/ar06-R4-cycle/routing.json	f5d80d3b17574a4e…
runs/e1/routing/ar06-R4-repeat/routing.json	aa1d17ff2e6101df…
runs/e1/routing/ar12-R3/routing.json	f2a658ef9b2435dd…
runs/e1/routing/compare.json	d9f855de4d799580…
runs/e1/routing/pn06-R3/routing.json	e677edd9ee5b1ed4…
runs/e1/smoke/ar3-b1/receipt.json	53c19121b2e16c9f…
runs/e1/smoke/ar3-b16/receipt.json	5b8d74f6b4541fda…
runs/e1/smoke/ar3-wiki/receipt.json	0415b5368f7efa20…
runs/e1/smoke/ar4r-b16/receipt.json	c16d6b1b31a4550a…
runs/e1/smoke/pn3-b16/receipt.json	de8521ba99a88472…
runs/e1/summary.json	8b24a01e1bc4060a…
runs/e1/wikitext-bos/ar06-R3/receipt.json	4e5dd12b362951db…

FILE	SHA-256
runs/e1/wikitext-bos/pn06-R3/receipt.json	04b3c2f182ba58fc…
runs/e1/wikitext/ar06-R1/receipt.json	6173e986df77414e…
runs/e1/wikitext/ar06-R2/receipt.json	efdae4b1eace91f3…
runs/e1/wikitext/ar06-R3/receipt.json	d2cf2424a0264a91…
runs/e1/wikitext/ar06-R4-cycle/receipt.json	96920bab5a302ba7…
runs/e1/wikitext/ar06-R4-repeat/receipt.json	dc984296a75ff423…
runs/e1/wikitext/ar06-R5-repeat/receipt.json	8dcad36e58a4da43…
runs/e1/wikitext/ar06-R6-repeat/receipt.json	0ebd92ea9438505e…
runs/e1/wikitext/ar12-R3/receipt.json	33a2b7384cb0f8cb…
runs/e1/wikitext/pn06-R1/receipt.json	6a047be56252927c…
runs/e1/wikitext/pn06-R2/receipt.json	08126b8914c56cb9…
runs/e1/wikitext/pn06-R3/receipt.json	42e36fd339e708a2…
runs/e1/wikitext/pn06-R4/receipt.json	adf2b693bb981059…
runs/e1/wikitext/pn06-R5/receipt.json	587ba588958b7c2e…
runs/e1/wikitext/pn06-R6/receipt.json	996dc500e01a702f…
runs/e2/referee.md	b1c877930141566e…
runs/e3/referee.md	0d19af74356eee0d…
runs/e3/referee/e3_referee.json	6025523b6a07da5b…
runs/e3/referee/e3_referee.py	a8d52d3ada846f18…
runs/e3/referee/sacct.txt	f7a3918f0a4ada77…
runs/f10/stage1c_routing_params-final.json	6e92c1b35cf578f6…
runs/s1a/ladder.json	a5b67c288286aaf7…
runs/s1a/referee.md	69b8ed65d5af704a…
runs/s1a/referee/s1a_recount.json	c48a8b441f2b7ee9…
runs/s1a/referee/s1a_recount.py	780996470ef91438…
runs/s1a/referee/s1a_replay.json	bc482cddce36edfe…
runs/s1a/referee/s1a_replay.py	2494ae77d76ca3c3…
runs/s1a/sacct.txt	77c07d408596c2bb…
runs/s1b/referee.md	139522b7727a1c27…
runs/s1b/referee/s1b_recount.json	59c321f3f146f927…
runs/s1b/referee/s1b_recount.py	86c596702128f2f4…
runs/s1c/data/finemath4p-s00/audit.json	9c0703a1a2eaa29a…
runs/s1c/eval/gsm8k-bos/C3-R3/receipt.json	872ed5b48ba3f4a4…
runs/s1c/eval/gsm8k-bos/C3e-R3/receipt.json	6387964a34c30cb1…
runs/s1c/eval/gsm8k-bos/L4-R4/receipt.json	b7181a5d09470b75…
runs/s1c/eval/gsm8k-bos/L4e-R4/receipt.json	f20fc6de626a8a35…
runs/s1c/eval/wikitext/C3-R3/receipt.json	cce86926cb9f91dc…
runs/s1c/eval/wikitext/C3e-R3/receipt.json	34ec385ee123d4bf…
runs/s1c/eval/wikitext/L4-R4/receipt.json	1f71d4487cde1040…
runs/s1c/eval/wikitext/L4e-R4/receipt.json	a0899abb01fe2bba…
runs/s1c/jobs.tsv	c9a7ced67f7f79fd…
runs/s1c/loop4.json	db17e30c2493da73…
runs/s1c/referee.md	6f6e285f7a74f9ca…
runs/s1c/referee/s1c_recount.json	092db789f149851f…
runs/s1c/referee/s1c_recount.py	5bd10b79fe096268…
runs/s1c/train/C3/receipt.json	e265d790a9a6aeed…
runs/s1c/train/C3e/receipt.json	e2606686c2a457d4…
runs/s1c/train/L4/receipt.json	dcebb14c694220df…
runs/s1c/train/L4e/receipt.json	6101dcf2146cd891…
runs/s2/analysis/depth.json	9b5984bc324186e2…
runs/s2/referee.md	e808a2cb93495edf…
runs/s4/referee.md	59f1e8f93495abec…
runs/s4/s4.json	a2084a2b56c7af79…
tools/leak_patterns.txt	2a9c5ba8de603107…