WHY DO EFFICIENT ARCHITECTURES LEARN EXPENSIVE REASONING? TRAJECTORY SELECTION UNDER OUTCOME-REWARD RL

Anonymous authors
Paper under double-blind review

ABSTRACT

Hybrid architectures that interleave linear-attention layers with a few softmax-attention layers are now post-trained with reinforcement learning from verifiable rewards, yet the architecture is still chosen on pre-training metrics. We ask a narrow question: *when the reward checks only correctness, why do some efficient models stably adopt long, explicit state-tracking trajectories, and is that cost a necessity of the architecture or a choice of the training process?* We pre-train parameter-matched 126M and 325M models (full attention, sliding window, pure Gated DeltaNet, GDN/KDA/GLA hybrids, MoE variants) on identical data, cold-start them with SFT data that contains both a terse and a state-summary trajectory style, and run identical GRPO/DAPO recipes on verifiable working-memory tasks. The GDN 7:1, KDA 3:1, GLA 3:1 and MoE+GDN hybrids switch to full state summaries on every problem ($2.7\times$ tokens, $\approx 3\times$ measured inference time) while full attention and GDN 3:1 stay terse; on programs with branches even full attention switches. Three controlled designs separate selection from necessity. Independently trained terse policies match the long policy for KDA 3:1 (-0.7 pp, 95% CI $[-2.7, +1.2]$) and beat it for GDN 3:1 on natural-language problems ($+3.9$ pp), but cost 5.2 pp for GDN 7:1 at 325M and 12–16 pp at 126M. Prefix edits show that every long policy *reads* its summaries, so within-policy dependence does not imply necessity. Mechanistically, the state style is *less* accurate on average, but its advantage inside the mixed-outcome groups that carry the gradient turns positive and is self-reinforcing; stratifying advantages within (prompt, style) abolishes the switch in 9/9 runs at the accuracy of the forbidden path. The same statistic predicts *who* switches across initialisation seeds with identical pre-training loss and across difficulty bands. Generic length penalties lose 23–43 pp for an identifiable reason, and once the summary strategy is removed it is not rediscovered. RL-selected long trajectories reverse the hybrids' deployment-cost advantage ($0.6\times \rightarrow 1.6$–$2.0\times$).

1 INTRODUCTION

Hybrid language models combine recurrent token mixers with a minority of softmax-attention layers (Lieber et al., 2024; MiniMax, 2025; NVIDIA, 2025; Kimi Team, 2025b). Recurrent layers reduce costs that grow with context length; full-attention layers still incur those costs. Comparisons often measure language modelling, recall and computation (Wang et al., 2025). Outcome-reward RL (Shao et al., 2024; DeepSeek-AI, 2025; Yu et al., 2025) can change the length and format of a model's solutions. We test whether the resulting policies retain their inference-cost advantage.

On synthetic tasks that require tracking several quantities, some models learn to append a full state summary to each reasoning step. On the held-out grid, the two switching 325M-class hybrids produce 2.4–$2.7\times$ as many tokens as separately trained terse counterparts. On a separate H100 profiling grid at batch 256, they take 2.8–$3.1\times$ as long per problem as their terse counterparts and are slower than full attention. Whether the extra computation improves accuracy depends on the base model and task.

A trained policy may rely on summaries even if another policy with the same architecture can solve the task tersely. We compare separately trained policies to measure attainable accuracy–cost trade-offs, then edit frozen long policies to test their reliance on summaries.

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

Contributions.

1. *Trajectory selection and cost* (§3). Several recurrent hybrids switch to state summaries under a shared RL recipe; full attention also switches on branching programs. Initialisation and training difficulty change this behaviour even when pre-training validation losses are nearly equal.
2. *Policy dependence and alternative paths* (§4). Paired comparisons identify an accuracy–cost trade-off for the main GDN 7:1 base, a small but statistically inconclusive difference for KDA 3:1, and higher observed mean accuracy after terse training for GDN 3:1 on narrative programs. Prefix edits distinguish these between-policy results from dependence within an existing policy.
3. *Mechanism and intervention* (§5–6). Style-conditional advantages in groups with both correct and incorrect answers reveal patterns that overall style accuracy misses. Normalising advantages within style prevents switching in nine tested runs. We also test how the style mix in supervised fine-tuning (SFT), reward changes and length penalties affect switching.

2 Experimental setup

Architectures. Within each scale, models share the tokenizer (16k BPE with digit splitting), data order, optimiser and schedule. Dense models have approximately matched mixer parameter counts. The **126M-class** grid (16 layers, d=768) includes A0 full attention, A1 sliding-window attention (window 512), A2 pure Gated DeltaNet (GDN) (Yang et al., 2025), A3a/A3b/A3c GDN hybrids with $1/8$, $1/4$, $1/2$ softmax-attention layers (GDN 7:1, GDN 3:1, GDN 1:1), A4 KDA 3:1 (Kimi Team, 2025b), and A5 Gated Linear Attention (GLA 3:1) (Yang et al., 2024). We also test mixture-of-experts (MoE) variants M0 (attention) and M1 (GDN 3:1); their total parameter counts are not matched to the dense grid. The **325M-class** grid (24 layers, d=1024, 16 attention heads) includes A0–A4. Hybrid attention layers are evenly spaced, including the final layer; linear-attention kernels come from FLA (Yang & Zhang, 2024). The main grid uses one pre-training initialisation per architecture ("base seed 0"). At 325M, A3a and A4 each have two more initialisations ("base seeds 1 and 2") with the same token stream and order. Base seeds differ from the RL seeds used to repeat post-training.

Data. The reported pre-training budgets are 2.49B tokens for the smaller models and 8.4B for the larger models, using FineWeb-Edu (62–63%) (Penedo et al., 2024), FineMath (13%) (Ben Allal et al., 2025), and synthetic task documents (24–25%). The released generators sample both terse and state-summary solutions for style-enabled tasks, including during pre-training. The smaller-model stream had an unintended synthetic-heavy tail (Appendix E); the larger-model stream uses corrected interleaving. We use the larger grid for primary comparisons. The smaller grid has a different data mixture, so cross-scale differences cannot be attributed to model size alone.

Tasks and styles. Tasks are program-generated, automatically scored and adjustable in difficulty. VARTRACK executes N assignments over 3–6 variables and asks for one variable; STORY presents similar programs as narratives. BRTRACK adds `if` statements and bounded `repeat` loops; difficulty counts top-level program lines before loop expansion. Other tasks are SORT, LOOKUP and ARITH. On tasks with two solution formats, a **terse** trajectory gives one computed line per step; a **state** trajectory also gives the variable state or remaining list. For example, `a = 4 + 3 = 7` is terse; adding `| a=7, b=3, c=1` makes the same step state-style. A rule-based detector measures the *restatement rate*: the fraction of reasoning lines with a summary. For style-conditional RL statistics, a rollout is state-style when this rate exceeds 0.5; FORBID instead penalises any detected summary (rate > 0).

Cold-start SFT. The standard set contains 40k examples, sampled with task probabilities 0.4 for VARTRACK and 0.2 each for SORT, LOOKUP and ARITH. Styles are sampled 50/50 on style-enabled tasks. E4 varies their state-style fraction (0/25/50/75/100%) while preserving prompts and answers. For STORY and BRTRACK, 20k task-specific examples are added to the common set (60k before length filtering).

RL recipe. We use group relative policy optimisation (GRPO) (Shao et al., 2024) with DAPO-inspired components (Yu et al., 2025): asymmetric clipping (0.2/0.28), filtering of all-equal reward groups, and token-level loss aggregation, without a KL penalty. Each of 300 rollout iterations samples 64 prompts with eight responses each; the learning rate is 3×10^{-6}, selected on A0 and shared

across architectures. The task reward is 1 for a correct, terminated response and 0 otherwise. The implementation uses two optimiser minibatches and a conditional entropy bonus (Appendix A). The main RL task mix is VARTRACK/SORT, sampled equally. The default VARTRACK band is 16–32 statements (12–24 for BRTRACK); harder bands are examined in §3.1. We call a policy that produces summaries after this *standard recipe* a *long policy*. Two interventions test other training paths: FORBID assigns zero reward to responses with detected summaries; *style-stratified normalisation* (SSN, E6) centres and scales rewards within each (prompt, style) subset. Both keep the same answer scorer but change the RL objective.

Evaluation and statistics. Final accuracy comparisons use held-out problems generated with fresh random seeds and greedy decoding, with paired per-item records. The VARTRACK/STORY grid contains 80 items at each of 16, 20, 24, 28, 32 and 40 statements; Appendix D describes the other grids and uncertainty estimates. Unless stated otherwise, reported 95% CIs bootstrap item–RL-seed pairs (three RL seeds in the primary comparisons, fewer in specified cells). Repeated prompts and shared base initialisations limit what these intervals say about between-run uncertainty, so we also report seed-level values. Cost is generation wall-clock time per problem on one H100 at batch 1, 32 and 256 using shared decoding code. The non-inferiority criterion is a one-sided 95% lower bound above -1 pp; the cost criterion requires at least 20% less time per problem at batch 256. Appendix B distinguishes the internal follow-up plan from later recorded predictions.

3 WHICH MODELS LEARN LONG TRAJECTORIES, AND AT WHAT COST?

Under the standard recipe. After 300 rollout iterations at the 325M scale (three RL seeds each), GDN 7:1 and KDA 3:1 append state summaries on nearly every held-out VARTRACK item (greedy restatement rate > 0.99; 817–842 tokens per problem). The other main-grid architectures stay terse in the monitoring evaluations. Table 2 gives final held-out results for the paired comparisons. At 126M, GDN 7:1, KDA 3:1, GLA 3:1 and MoE+GDN also switch, while full attention, SWA, pure GDN, GDN 3:1, GDN 1:1 and attention-MoE stay terse.

On narrative STORY problems, GDN 7:1 and GDN 3:1 switch (restatement > 0.99 at iteration 300), while full attention stays terse. On BRTRACK, which adds branches and loops, all three tested architectures switch, including full attention (restatement 0.95–1.0 by iteration 150).

Cost. At batch 256, terse hybrids need 12–14 ms per problem against 21 ms for full attention ($0.6\times$); the same hybrids with RL-selected state trajectories need 34–42 ms (1.6–2.0$\times$ full attention; Figure 1, right; Table 1). At batch 1 the gap is larger for GDN 7:1: 5.3 s with summaries versus 1.1 s terse and 1.4 s for full attention. In this profiling setup, the longer trajectories erase the hybrids' time-per-problem advantage. Section 4 tests whether the extra time improves accuracy.

Table 1: Generation time per problem on one H100 with shared greedy decoding (base seed 0, RL seed 0). Batch 1: one 16-item run; batches 32/256: median of three 256-item runs. Profiling uses 16–32 statements; tokens use batch 256. Accuracy averages three RL seeds on a separate 480-item grid (16–40 statements).

policy	tokens/problem	batch 1 (s)	batch 32 (s)	batch 256 (s)	accuracy
full attn, terse (standard)	276	1.41	0.059	0.0208	.831
GDN 3:1, terse (standard)	276	1.30	0.049	0.0135	.689
GDN 7:1, state (standard)	763	5.31	0.117	0.0341	.634
GDN 7:1, terse (FORBID)	276	1.09	0.043	0.0120	.582
KDA 3:1, state (standard)	763	3.17	0.139	0.0417	.856
KDA 3:1, terse (FORBID)	277	1.37	0.052	0.0133	.849

3.1 SWITCHING DEPENDS ON INITIALISATION AND TASK DIFFICULTY

Two further initialisation seeds. Across three initialisations of each hybrid, the last recorded FineWeb validation losses differ by at most 0.0033 (GDN 7:1: 2.461–2.462; KDA 3:1: 2.441–2.444). Yet after the same SFT, held-out VARTRACK accuracy is 0.48 / 0.79 / 0.53 for GDN 7:1 and 0.76 /

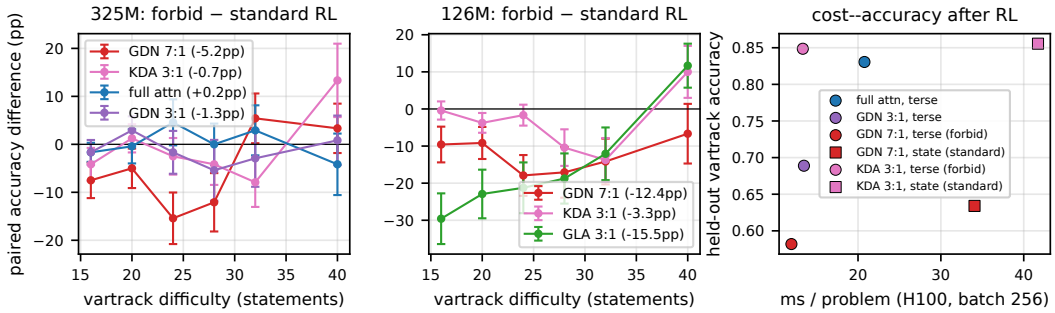

Figure 1: **Left, middle:** paired held-out accuracy difference (FORBID minus standard RL) by program length at 325M and 126M. Error bars are normal-approximation 95% intervals (± 1.96 SE) over item–seed pairs; the legend gives the overall difference. The accuracy difference becomes positive at the longest programs for several hybrids, while it remains negative for GDN 7:1 at 126M. **Right:** time per problem on one H100 (batch 256, greedy) versus held-out accuracy on a separate task grid (Table 1). Squares show RL-selected state trajectories. The same hybrids trained with FORBID (same-colour circles) use 32–35% of their generation time.

0.95 / 0.95 for KDA 3:1 (base seeds 0 / 1 / 2). Similar language-modelling losses therefore do not imply similar task accuracy.

Under standard RL, the three bases with higher SFT accuracy (GDN 7:1 seed 1; KDA 3:1 seeds 1 and 2) stay terse throughout the 300-iteration budget.

In these runs, the share of state-style training rollouts falls from about one quarter to low single-digit percentages. The resulting terse policies reach 0.94–0.96 held-out accuracy, above the seed-0 long policies (0.634 for GDN 7:1, 0.856 for KDA 3:1). For GDN 7:1 seed 2, SFT accuracy is close to seed 0 and state-style conditional advantage is positive in some windows (§5). Its greedy policy nevertheless remains terse at iteration 300, with 0.689 held-out accuracy. The FORBID policy has 2.0 pp lower observed accuracy than standard RL ($[-4.2, +0.2]$). The seed-1 bases also stay terse on 24–40 and 40–64 training bands; the seed-1 GDN 7:1 tracks 64-statement programs tersely at 0.85.

A harder training band on the switching bases. Training the seed-0 bases on 24–40-statement programs with the standard recipe, everything else unchanged, also produces no switch (two RL seeds each; restatement ≤ 0.05). For GDN 7:1, the hard-band terse policy is less accurate at 24 and 28 statements (0.66 and 0.49, versus 0.89 and 0.60 for the default-band long policy). At 32 and 40 statements, it is more accurate (0.38 and 0.25 versus 0.31 and 0.11). Across the full held-out grid, the hard-band policy is 4.9 pp less accurate ($[-7.7, -2.1]$). For KDA 3:1, the hard-band terse policy is $+2.3$ pp $[-0.1, +4.7]$ above the default-band long policy (Table 8).

Switching varies with the starting model and training difficulty. Section 5 examines the corresponding style-conditional advantages. Some runs have positive advantage windows yet remain terse, so a positive value alone does not predict a switch.

4 Policy dependence and alternative training paths

4.1 Between policies: independently trained short paths (E1, E4)

For each comparison, we train standard-reward and FORBID policies from the same SFT checkpoint, pair them by RL seed, and evaluate them on the same held-out items (Table 2). FORBID changes the reward while keeping the answer scorer. A second short path uses a 0%-summary SFT fraction followed by the standard reward (§6). These are two tested paths to terse output; neither can establish whether the architecture itself needs summaries.

KDA 3:1 (base seed 0). FORBID gives -0.7 pp $[-2.7, +1.3]$ and uses approximately 63% fewer response tokens. The 0%-summary SFT path gives $+0.1$ pp $[-2.0, +2.2]$. Terse control architectures give $+0.2$ pp (full attention) and -1.3 pp (GDN 3:1). For KDA 3:1, the one-sided 95% lower bound

Table 2: **E1: accuracy and response length of independently trained policies.** Standard and FORBID accuracies are listed by RL seed. Differences (FORBID − standard) and 95% bootstrap CIs pool matched item–seed pairs and condition on the evaluated policies; a CI spanning zero does not establish non-inferiority. Style is the standard-RL greedy output; tokens are mean response lengths.

scale / task	base	style	standard acc.	FORBID acc.	tokens	FORBID − std. (pp)
325M, VARTRACK	GDN 7:1	state	.625/.644/.633	.579/.575/.592	825→344	$\mathbf{-5.2}$ [−7.3, −3.1]
	KDA 3:1	state	.844/.852/.871	.848/.844/.854	839→309	−0.7 [−2.7, +1.3]
	full attn	terse	.833/.833/.825	.825/.817/.856	309→311	+0.2 [−1.7, +2.2]
	GDN 3:1	terse	.673/.700/.694	.660/.675/.692	297→307	−1.3 [−3.3, +0.6]
126M, VARTRACK	GDN 7:1	state	.838/.815/.825	.688/.715/.702	844→310	$\mathbf{-12.4}$ [−14.8, −10.1]
	GLA 3:1	state	.523/.525/.500	.344/.363/.377	813→308	$\mathbf{-15.5}$ [−18.2, −12.6]
	KDA 3:1	state	.900/.904/.908	.877/.871/.865	841→309	$\mathbf{-3.3}$ [−5.2, −1.5]
325M, BRTRACK	GDN 7:1	state	.642/.667	.615/.629	807→327	$\mathbf{-3.2}$ [−5.8, −0.7]
	KDA 3:1	state	.825/.810	.815/.823	817→330	+0.1 [−2.4, +2.6]
	full attn	state	.756/.765	.756/.752	803→337	−0.6 [−3.4, +2.4]
325M, STORY	GDN 7:1	state	.790/.810/.788	.773/.767/.771	911→330	$\mathbf{-2.6}$ [−4.9, −0.2]
	GDN 3:1	state	.883/.892	.917/.935	922→334	$\mathbf{+3.9}$ [+1.8, +6.0]
325M, other seeds	GDN 7:1 s1	terse	.946/.940	.946/.952	312→311	+0.6 [−0.8, +2.0]
	GDN 7:1 s2	terse	.690/.688	.660/.677	320→334	−2.0 [−4.2, +0.2]
	KDA 3:1 s1	terse	.963/.965	.948/.954	310→309	−1.3 [−2.5, +0.0]
	KDA 3:1 s2	terse	.960/.963	.965/.954	310→309	−0.2 [−1.4, +0.9]

is -2.36 pp, below the specified -1 pp threshold. The mean accuracy decrease is small, but **the non-inferiority test fails**. A CI spanning zero does not establish equivalence.

GDN 7:1 (base seed 0). FORBID gives -5.2 pp $[-7.3, -3.1]$ (mean accuracies 0.634 vs 0.582); the 0%-summary SFT difference averages -4.0 pp over three RL seeds. The FORBID differences are approximately $-8/-5/-15/-12$ pp at 16/20/24/28 statements and $+5/+3$ pp at 32/40 (Figure 1). The long policy helps most at intermediate lengths. The sign reverses at 32 and 40 statements; accumulated errors are one possible explanation, not a mechanism established here.

Scale. At 126M the corresponding differences are -12.4 pp for GDN 7:1, -15.5 pp for GLA 3:1 and -3.3 pp for KDA 3:1. The larger gaps are consistent with a greater benefit from summaries in these bases, although scale and pre-training mixture differ together.

Other tasks and bases (Table 2). On STORY, FORBID changes accuracy by -2.6 pp for GDN 7:1 and $+3.9$ pp for GDN 3:1; the latter interval excludes zero under the pooled analysis. On BRTRACK, differences are -3.2 pp (GDN 7:1), $+0.1$ pp (KDA 3:1) and -0.6 pp (full attention). Full attention therefore switches in both RL seeds without a clearly resolved accuracy benefit. The additional base-seed comparisons are reported separately in Table 2.

4.2 WITHIN A POLICY: PREFIX INTERVENTIONS ON FROZEN LONG POLICIES (E2)

We edit the first half of each frozen long policy's greedy prefix, retokenise it, rebuild caches, and continue generation. Every displayed cell contains 240 held-out items meeting the eligibility rules: at least eight reasoning lines and detected summaries throughout that half. Conditions are *full* (unedited prefix); *oracle* (replace only the final prefix summary with the true state; VARTRACK only); *corrupt* (alter summary values); *swap* (shift summaries cyclically by one line); *filler* (replace numeric characters with `x`); *strip* (remove summaries); *last only* (keep the last reasoning line with its summary); and *pc only* (keep that line without its summary). The final condition keeps the computed line, not just a program-counter token (Table 3).

All differences below use the reconstructed *full* condition as reference. On BRTRACK, even reconstructing the unedited prefix lowers accuracy by 8.8–12.5 pp relative to the original completions (Appendix A). Across the displayed cells, corruption lowers continuation accuracy by 10–65 pp and one-line summary shifts by 3–10 pp. Correcting only the final summary changes accuracy by at most 1.3 pp; we do not correct the entire prefix. At 24 statements, stripping summaries costs about 3 pp for KDA 3:1 and 15 pp for GDN 7:1; on BRTRACK, it costs 5–11 pp, including for full attention. The

Table 3: **E2: prefix edits on frozen long policies.** Continuation accuracy uses 240 eligible held-out items per cell. Full is the reconstructed, unedited prefix; all edits are compared with this control, which differs from the original completion on BRTRACK. Oracle corrects only the final prefix summary (VARTRACK only); swap shifts summaries by one line. Table 2 tests separately trained alternatives.

task, difficulty	policy	full	oracle	corrupt	swap	filler	strip	last only	pc only
VARTRACK, 24	GDN 7:1	.883	.896	.521	.796	.408	.733	.571	.546
VARTRACK, 24	KDA 3:1	.921	.929	.575	.850	.321	.892	.767	.771
VARTRACK, 32	GDN 7:1	.388	.375	.288	.358	.238	.346	.325	.321
BRTRACK, 20	full attn	.667	–	.054	.563	.088	.558	.375	.308
BRTRACK, 20	GDN 7:1	.563	–	.050	.496	.113	.471	.317	.292
BRTRACK, 20	KDA 3:1	.771	–	.117	.729	.242	.717	.667	.650

edits may shift the prefix distribution. They show sensitivity of these frozen policies to summary edits, **not an architectural need for summaries**.

5 GRPO ADVANTAGES BY RESPONSE STYLE

GRPO compares eight responses to the same prompt. A response can score above its prompt group even if its style is less accurate across prompts. For prompt q, the standard recipe assigns response i the advantage

$$A_{qi} = \frac{r_{qi} - \bar{r}_q}{\sigma_q + 10^{-6}}, \tag{1}$$

where $\bar{r}_q$ and σ_q are the population mean and standard deviation of the eight rewards. A group is *retained* when its rewards differ. Under the binary standard reward, all-equal groups have zero advantage and are excluded from updates. We track each style's frequency, accuracy and mean advantage among retained-group samples. Mean advantage alone does not determine the update to a style's probability; token weighting, clipping, shared parameters, entropy regularisation and optimiser dynamics also matter.

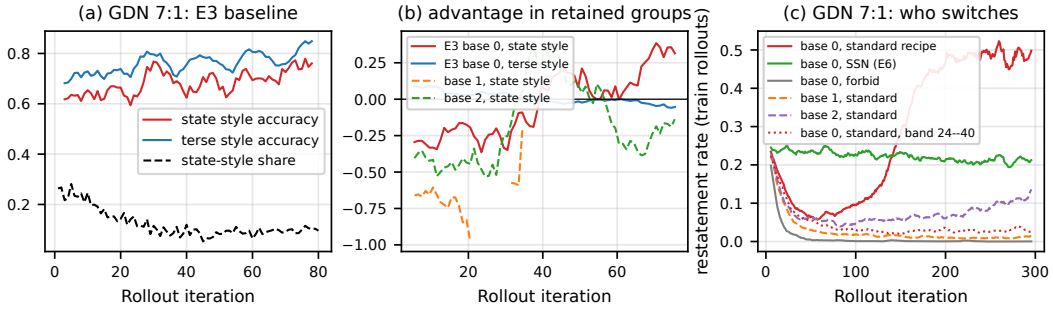

Figure 2: **Style diagnostics for GDN 7:1 (325M).** (a) Style accuracy and state-style share in the 80-iteration E3 baseline, pooled over the VARTRACK/SORT training mixture. (b) Mean retained-group advantages by style in that baseline and in the first 80 iterations of the standard runs from base seeds 1 and 2. (c) Restatement rates in separate 300-iteration runs. About half the prompts are SORT, so a rate near 0.5 can mean near-complete switching on VARTRACK. Accuracy in (a) uses five-iteration moving averages; style share is unsmoothed. Panels (b,c) use ten-iteration averages, with gaps where statistics are missing. All runs use RL seed 0.

Style accuracy and advantage can diverge. In the E3 baseline, state-style responses are less accurate on average than terse responses over the early training mixture, and their sampling share initially falls (Figure 2a). Yet their mean retained-group advantage changes from -0.252 over iterations 1–20 to $+0.119$ over iterations 41–60 (Figure 2b). The logs pool VARTRACK and SORT, so they do not isolate a task-specific effect.

For illustration, consider three hypothetical prompts with eight responses each: eight correct terse responses; four correct state-style and four incorrect terse responses; and eight incorrect state-style responses. State-style accuracy is $4/12$, versus $8/12$ for terse responses. Only the second group is retained, giving each state-style response advantage approximately $+1$.

Starting model and difficulty change the pattern. The high-accuracy base-seed-1 models stay terse and show negative early state-style retained-group advantages: -0.73 for GDN 7:1 and -1.09 for KDA 3:1 over the first 30 iterations. The borderline GDN 7:1 base seed 2 has positive windows but stays terse, as do the tested harder-band runs. Positive conditional advantage alone does not predict a switch. The STORY runs also show that overall style accuracy is an incomplete diagnostic.

Recipe ablations (E3, E5). All 80-iteration E3 VARTRACK runs stay terse. On STORY, removing group standard-deviation normalisation (Liu et al., 2025), using symmetric clipping, or applying the filtering ablation yields summary-dominated output. The filtering ablation still drops zero-advantage samples; it does not retain all-equal groups in the loss denominator. In longer VARTRACK runs, no standard-deviation normalisation, the combined "plain GRPO" ablation (symmetric clipping, sequence-level loss and the no-filter flag), and sequence-level loss each switch. Both the standard and sequence-level-loss runs have switched by the iteration-150 evaluation; their relative switch times are unknown. Final accuracy differs from standard by $+2.4$ pp, -1.3 pp and $+0.3$ pp, respectively (Table 4). These runs do not isolate a single recipe component as the cause of switching.

Within-style normalisation (E6). SSN replaces Equation 1 with

$$A_{qi}^{\mathrm{SSN}} = \begin{cases} (r_{qi} - \bar{r}_{q,s_i})/(\sigma_{q,s_i} + 10^{-6}), & n_{q,s_i} \geq 2, \\ 0, & n_{q,s_i} < 2, \end{cases} \tag{2}$$

where s_i is the detected style and n_{q,s_i} counts rollouts of that style. The mean advantage within each style subset is zero. This removes between-style mean advantages, but also changes scaling and gradient sparsity. Task reward, prompt sampling, rollout-group size, learning rate, clipping and loss aggregation are unchanged; generation microbatch sizes vary with memory requirements.

All nine tested SSN runs finish with terse greedy policies: three RL seeds each for GDN 7:1 and KDA 3:1 at 325M, one smaller GDN 7:1 run, and one STORY run for each of GDN 7:1 and GDN 3:1. At 325M, mean VARTRACK accuracies are 0.590 and 0.840, compared with 0.582 and 0.849 under FORBID. On STORY, differences from standard RL are -2.1 pp for GDN 7:1 and $+4.2$ pp for GDN 3:1 (Table 9). Across settings, configuration-level mean accuracies lie within 1 pp of the matched FORBID controls, without establishing equivalence. Because SSN also changes scaling and gradient sparsity, these runs do not isolate a unique cause of switching.

6 CONTROLLING RESPONSE STYLE

Table 4: **Recipe variants for GDN 7:1 (325M, held-out VARTRACK).** Differences use standard RL as reference except in the matched 600-iteration comparison and E7. One-run recipe ablations are compared with three standard runs; their CIs pool item–seed pairs. E7 uses its matched seed-0 baseline.

variant	acc.	tokens	final style	paired diff. (pp)
standard GRPO+DAPO (3 seeds)	.634	825	state	–
plain GRPO	.621	811	state	-1.3 $[-3.1, +0.3]$
no std-normalisation	.658	837	state	$+2.4$ $[+0.4, +4.4]$
sequence-level loss	.638	850	state	$+0.3$ $[-1.5, +2.1]$
length penalty 0.05 / 0.1 / 0.2	.206 / .260 / .244	217 / 228 / 296	terse	-42.8 / -37.4 / -39.0
stage-2 penalty (100 iterations)	.406	276	terse	-22.8 $[-25.3, -20.3]$
0% state-style SFT prior (3 seeds)	.594	338	terse	-4.0
FORBID (3 seeds)	.582	344	terse	-5.2 $[-7.3, -3.1]$
SSN style-stratified (3 seeds)	.590	335	terse	-4.4 $[-6.5, -2.4]$
600 iterations: FORBID vs standard	.527 vs .646	320	terse	-11.9 $[-16.3, -7.7]$
Release, 150 iterations (E7)	.615	320	terse	-1.0 $[-4.6, +2.7]$

SFT style changes the selected path (E4). Changing the fraction of state-style examples while keeping prompts and answers fixed changes the final policy (Figure 3; Table 7). With no state-style SFT examples, neither hybrid switches within 300 iterations (GDN 7:1: about -4 pp; KDA 3:1: $+0.1$ pp versus the standard 50% fraction). Fractions of 25–75% lead to switching in the tested hybrid runs. A 100% fraction produces state-style VARTRACK trajectories but reduces SORT accuracy from 0.831 to 0.470 for GDN 7:1 and from 0.829 to 0.477 for KDA 3:1. Repeated list summaries or limited exposure to terse trajectories may contribute; this experiment does not separate them. Both styles also occur in pre-training, so the SFT result does not show that an absent SFT style is unreachable. Full attention switches with the 75% fraction (-2.2 pp).

Persistence after restoring the reward (E7). Starting from each terse FORBID policy, we restore the standard reward for 150 iterations (one run per architecture). Both greedy policies remain terse. GDN 7:1 samples 37 state-style rollouts out of 76,800 (0.048%); KDA 3:1 samples none. Held-out accuracy reaches 0.615 for GDN 7:1 ($+3.5$ pp over its FORBID checkpoint; -1.0 pp versus its matched long policy) and 0.865 for KDA 3:1 ($+1.7$ pp and $+2.1$ pp, respectively). Summaries remain rare over this budget, but the 37 GDN 7:1 samples show that they remain in its sampling support.

More iterations in one run. Extending the main GDN 7:1 FORBID run from 300 to 600 iterations lowers held-out VARTRACK accuracy from 0.579 to 0.527. Its matched long policy improves from 0.625 to 0.646, increasing the gap to -11.9 pp. More iterations alone did not improve this terse policy; this one run does not rule out other schedules.

Length-penalty baselines (E5). Motivated by reward-based length control (Kimi Team, 2025a; Aggarwal & Welleck, 2025), we test our own correctness-gated penalty,

$$r_\lambda(y) = c(y)\left(1 - \lambda \frac{L(y)}{1536}\right), \tag{3}$$

where $c(y)$ indicates a correct, terminated response and $L(y)$ is its token count. For GDN 7:1, $\lambda = 0.05, 0.1, 0.2$ lowers accuracy by 43, 37 and 39 pp. Applying $\lambda = 0.1$ for 100 more iterations from a long-policy checkpoint lowers accuracy from 0.625 to 0.406 (-21.9 pp). This second-stage run differs from concise-reasoning RL (Fatemi et al., 2025) in problem selection. In an all-correct group whose responses differ in length, reward standardisation largely cancels the coefficient λ (up to numerical stabilisation), leaving a preference for shorter outputs. Unequal length rewards also make such groups eligible for updates. A small coefficient can therefore yield a strong length signal. These results cover the tested penalties, not all length-control methods.

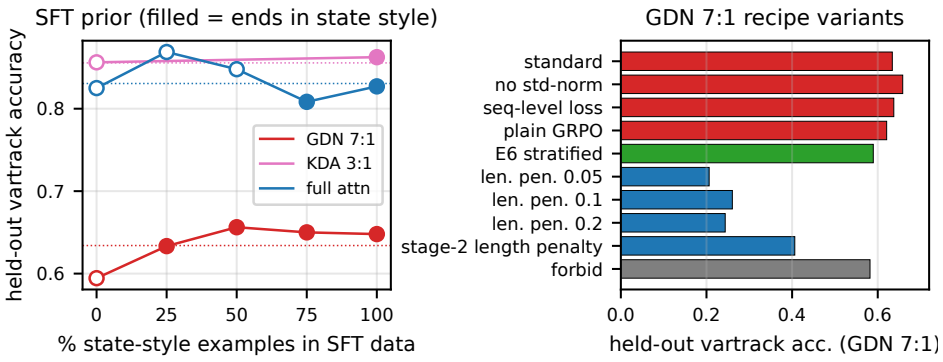

Figure 3: **Left (E4):** held-out accuracy after 300 rollout iterations by SFT state-style fraction. Prompts, answers and the configured recipe match across fractions; filled markers indicate final state style, and dotted lines show means for the 50% SFT fraction. **Right (E5/E6):** GDN 7:1 recipe variants. Standard/SSN/FORBID bars average three RL seeds; other bars show one run. Red indicates final state style, blue terse. Length penalties lower accuracy; SSN (green) is close in point estimate to FORBID (grey).

7 Related work

Efficient architectures under post-training. Liu et al. (2026) study RL with sliding-window attention; Kimi Team (2025b) report RL gains for KDA hybrids relative to their full-attention baseline. Our independently trained small models measure the accuracy and generation cost of the trajectories RL selects; we do not extrapolate our results to those production models.

Wang et al. (2025) compare hybrid ratios in language modelling and recall. Afendulev et al. (2026) examine attention and recurrence in trained hybrids, and Ye et al. (2026) derive expressiveness separations under stated assumptions. Bounds for the SSMs studied by Merrill et al. (2024) do not directly apply to our GDN or KDA policies. Tool access offers another path to length generalisation in SSMs (Malach et al., 2025); our tests change visible reasoning tokens. Evaluation also matters: Kostelec & Guo (2026) find differences between likelihood-based and generation-based evaluation of distilled hybrids, while Zhou et al. (2026) report impaired long-context recall after CoT fine-tuning.

Written intermediate state and reasoning length. Chain-of-thought, scratchpads and Self-Notes show that written intermediate computation can improve reasoning (Wei et al., 2022; Nye et al., 2021; Lanchantin et al., 2023). We ask when outcome-reward RL selects explicit summaries and how other training paths change accuracy and cost. Liu et al. (2025) analyse length and difficulty biases in GRPO. Other work studies concise reasoning (Fatemi et al., 2025), explicit length control (Aggarwal & Welleck, 2025), and reward shaping for long reasoning (Yeo et al., 2025). We compare terse and summary-bearing trajectories using conditional advantages and within-style normalisation.

8 Discussion and limitations

Training paths and policy dependence. Training with FORBID tests whether a terse policy can be learned from a given SFT checkpoint. Prefix edits measure how a long policy's continuation changes when its summaries are altered. A drop after editing does not show that the architecture needs summaries; a successful terse policy does not show that a long policy can dispense with them. The accuracy cost of terse output varies across bases, tasks and program lengths.

Interpreting generation cost. Complete-response time depends on both per-token execution and response length. Our terse and long hybrids share an architecture but follow different training paths, so their timing difference does not isolate per-token speed. Figure 1 reports timings alongside accuracies from a separate held-out grid; the two measurements do not use the same items. Time per problem also changes with batch size (Table 1). These results support comparing post-RL models by accuracy and measured response time under a stated workload and decoding setup.

Uncertainty across items and seeds. The primary intervals pool item–RL-seed pairs. An exploratory check that resamples each item jointly across seeds gives a 95% CI of $[-7.7, -2.8]$ pp for GDN 7:1 (FORBID minus standard) and $[-3.2, +1.9]$ pp for KDA 3:1 on VARTRACK. A seed-level interval for the STORY GDN 3:1 gain is $[-2.8, +10.5]$ pp, although its pooled interval is positive (Appendix D). These checks leave the main GDN 7:1 difference below zero, but the STORY gain is less certain across RL seeds. They still condition on the tested pre-training and SFT checkpoints.

We test three pre-training initialisations for two architectures and one for the others, with one SFT checkpoint per base and style fraction. Large seed effects limit architecture-level rankings, and different pre-training mixtures confound scale comparisons. All tasks are synthetic, including STORY, and timing uses one GPU type and implementation. Prefix reconstruction changes the generation path; edits may also shift the input distribution. SSN changes scaling and gradient sparsity along with between-style mean advantages, and the length penalties cover a narrow reward family. Generalisation to production models or mathematical reasoning remains untested.

Outcome-reward RL can select summaries that erase a hybrid model's speed advantage on these tasks. Style-conditional advantages reveal training signals that overall style accuracy misses, and within-style normalisation prevents switching in the tested runs. Efficiency therefore needs to be evaluated for the trained policy, including its measured time, response length and accuracy.

Reproducibility statement

Sections 2 and Appendix A specify the models and training recipe; Appendix D specifies evaluation and uncertainty estimates. The anonymous supplement contains code, configurations, the tokenizer, training logs, SFT hashes, per-item evaluations, prefix and timing outputs, and scripts to recompute the reported aggregates and figures. It does not include checkpoints, full SFT datasets or pre-tokenised training streams, so it supports checking the reported records but not independent retraining or inference. Checkpoints and streams are planned for camera-ready release. Appendices B and E explain the internal plan and data provenance.

AI use statement

Generative AI tools assisted with initial manuscript drafting, translation, and language editing. During revision, these tools also supported literature verification and consistency checks across the manuscript, implementation, and saved experimental results. References and numerical claims were verified against primary publications and project artifacts. AI tools were additionally used to monitor experimental runs, conduct scheduled checks of the server cluster, and assist with debugging the experiments in conjunction with command-line tools. The authors are responsible for the final text, claims, and research artifacts.

Ethics statement

We use program-generated tasks and public text corpora (FineWeb-Edu and FineMath), without human-subject experiments or new collection of personal data. Web-derived corpora may still contain personal information and biases. Results on our synthetic tasks do not establish reliability in high-stakes settings. Reuse of the corpora and software remains subject to their access conditions and licences.

References

Kirill Afendulev, Alexey Dontsov, Elena Tutubalina, and Anton Korznikov. What attention recalls and recurrence controls in hybrid language models. *arXiv preprint arXiv:2609.04434*, 2026. URL https://arxiv.org/abs/2609.04434. Accepted to Findings of EMNLP 2026.

Pranjal Aggarwal and Sean Welleck. L1: Controlling how long a reasoning model thinks with reinforcement learning. *arXiv preprint arXiv:2503.04697*, 2025. doi: 10.48550/arXiv.2503.04697. URL https://arxiv.org/abs/2503.04697.

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Clémentine Fourrier, Ben Burtenshaw, Hugo Larcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, Colin Raffel, Leandro von Werra, and Thomas Wolf. SmolLM2: When smol goes big – data-centric training of a small language model. *arXiv preprint arXiv:2502.02737*, 2025. doi: 10.48550/arXiv.2502.02737. URL https://arxiv.org/abs/2502.02737.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. doi: 10.48550/arXiv.2501.12948. URL https://arxiv.org/abs/2501.12948.

Mehdi Fatemi, Banafsheh Rafiee, Mingjie Tang, and Kartik Talamadupula. Concise reasoning via reinforcement learning. *arXiv preprint arXiv:2504.05185*, 2025. doi: 10.48550/arXiv.2504.05185. URL https://arxiv.org/abs/2504.05185.

Kimi Team. Kimi k1.5: Scaling reinforcement learning with LLMs. *arXiv preprint arXiv:2501.12599*, 2025a. doi: 10.48550/arXiv.2501.12599. URL https://arxiv.org/abs/2501.12599.

Kimi Team. Kimi Linear: An expressive, efficient attention architecture. *arXiv preprint arXiv:2510.26692*, 2025b. doi: 10.48550/arXiv.2510.26692. URL https://arxiv.org/abs/2510.26692.

Juan Gabriel Kostelec and Qinghai Guo. When perplexity lies: Generation-focused distillation of hybrid sequence models. *arXiv preprint arXiv:2603.26556*, 2026. doi: 10.48550/arXiv.2603.26556. URL https://arxiv.org/abs/2603.26556.

Jack Lanchantin, Shubham Toshniwal, Jason Weston, Arthur Szlam, and Sainbayar Sukhbaatar. Learning to reason and memorize with Self-Notes. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-0520. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/274d0146144643ee2459a602123c60ff-Abstract-Conference.html.

Opher Lieber, Barak Lenz, Hofit Bata, Gal Cohen, Jhonathan Osin, Itay Dalmedigos, Erez Safahi, Shaked Meirom, Yonatan Belinkov, Shai Shalev-Shwartz, Omri Abend, Raz Alon, Tomer Asida, Amir Bergman, Roman Glozman, Michael Gokhman, Avashalom Manevich, Nir Ratner, Noam Rozen, Erez Shwartz, Mor Zusman, and Yoav Shoham. Jamba: A hybrid Transformer-Mamba language model. *arXiv preprint arXiv:2403.19887*, 2024. doi: 10.48550/arXiv.2403.19887. URL https://arxiv.org/abs/2403.19887.

Kai Liu, Peijie Dong, Xinchen Xie, Jianfei Gao, Qipeng Guo, Xiaowen Chu, Shaoting Zhang, and Kai Chen. Architecture-aware reinforcement learning makes sliding-window attention competitive in math reasoning. *arXiv preprint arXiv:2606.11634*, 2026. doi: 10.48550/arXiv.2606.11634. URL https://arxiv.org/abs/2606.11634.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding R1-Zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025. doi: 10.48550/arXiv.2503.20783. URL https://arxiv.org/abs/2503.20783.

Eran Malach, Omid Saremi, Sinead Williamson, Arwen Bradley, Aryo Lotfi, Emmanuel Abbe, Josh Susskind, and Etai Littwin. To infinity and beyond: Tool-use unlocks length generalization in state space models. *arXiv preprint arXiv:2510.14826*, 2025. doi: 10.48550/arXiv.2510.14826. URL https://arxiv.org/abs/2510.14826.

William Merrill, Jackson Petty, and Ashish Sabharwal. The illusion of state in state-space models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 35492–35506. PMLR, 2024. URL https://proceedings.mlr.press/v235/merrill24a.html.

MiniMax. MiniMax-01: Scaling foundation models with lightning attention. *arXiv preprint arXiv:2501.08313*, 2025. doi: 10.48550/arXiv.2501.08313. URL https://arxiv.org/abs/2501.08313.

NVIDIA. Nemotron-H: A family of accurate and efficient hybrid Mamba-Transformer models. *arXiv preprint arXiv:2504.03624*, 2025. doi: 10.48550/arXiv.2504.03624. URL https://arxiv.org/abs/2504.03624.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. Show your work: Scratchpads for intermediate computation with language models. *arXiv preprint arXiv:2112.00114*, 2021. doi: 10.48550/arXiv.2112.00114. URL https://arxiv.org/abs/2112.00114.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The FineWeb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-0970. URL https://proceedings.nips.cc/paper_files/paper/2024/hash/370df50ccfdf8bde18f8f9c2d9151bda-Abstract-Datasets_and_Benchmarks_Track.html. Datasets and Benchmarks Track.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. doi: 10.48550/arXiv.2402.03300. URL https://arxiv.org/abs/2402.03300.

Dustin Wang, Rui-Jie Zhu, Steven Abreu, Yong Shan, Taylor Kergan, Yuqi Pan, Yuhong Chou, Zheng Li, Jibin Wu, Ge Zhang, Wenhao Huang, and Jason Eshraghian. A systematic analysis of hybrid linear attention. *arXiv preprint arXiv:2507.06457*, 2025. doi: 10.48550/arXiv.2507.06457. URL https://arxiv.org/abs/2507.06457.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.52202/068431-1800. URL https://proceedings.neurips.cc/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

Songlin Yang and Yu Zhang. FLA: A Triton-based library for hardware-efficient implementations of linear attention mechanism. Software, January 2024. URL https://github.com/fla-org/flash-linear-attention.

Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. Gated linear attention transformers with hardware-efficient training. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 56501–56523. PMLR, 2024. URL https://proceedings.mlr.press/v235/yang24ab.html.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving Mamba2 with delta rule. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=r8H7xhYPwz.

Xiaowei Ye, Xiaoyu He, Chao Liao, Chen Wu, and Pinyan Lu. A provable expressiveness hierarchy in hybrid linear-full attention. *arXiv preprint arXiv:2602.01763*, 2026. doi: 10.48550/arXiv.2602.01763. URL https://arxiv.org/abs/2602.01763.

Edward Yeo, Yuxuan Tong, Morry Niu, Graham Neubig, and Xiang Yue. Demystifying long chain-of-thought reasoning in LLMs. *arXiv preprint arXiv:2502.03373*, 2025. doi: 10.48550/arXiv.2502.03373. URL https://arxiv.org/abs/2502.03373.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-3775. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/a4277440d50f1f15d2cb4c14f7e0c0d2-Abstract-Conference.html.

Xinyu Zhou, Boyu Zhu, Yi Xu, Zhiwei Li, Yingfa Chen, Huiming Wang, and Zhijiang Guo. Attention amnesia in hybrid LLMs: When CoT fine-tuning breaks long-range recall, and how to fix it. *arXiv preprint arXiv:2606.11052*, 2026. doi: 10.48550/arXiv.2606.11052. URL https://arxiv.org/abs/2606.11052. Accepted to EMNLP 2026 (Main).

A Additional setup details

Model dimensions. The smaller dense grid uses 16 layers, hidden size 768, 12 attention heads and a SwiGLU feed-forward width of 2048. The larger grid uses 24 layers, hidden size 1024, 16 attention heads and feed-forward width 2816. GDN/KDA use 8 heads at the smaller scale and 10 at the larger scale, with key dimension 64 and value dimension 96. Hybrid attention indices are determined by evenly spacing the prescribed number of full-attention layers and including the final layer. The saved configurations are included in the supplement. The scale labels are approximate; the actual parameter counts differ, particularly for larger KDA models. MoE models have 16 routed experts, top-3 routing and one shared expert per MoE layer, expert width 512, and two initial dense layers. Their active and total parameter budgets must be distinguished.

Table 5: Recorded parameter counts (millions). Dense counts are total/active. MoE counts distinguish all stored parameters from those active for a token; both include embeddings. The scale labels are approximate.

scale	architecture	total (M)	active (M)
126M	A0	125.857	125.857
126M	A1	125.857	125.857
126M	A2	129.313	129.313
126M	A3a	128.881	128.881
126M	A3b	128.449	128.449
126M	A3c	127.585	127.585
126M	A4	124.557	124.557
126M	A5	126.159	126.159
126M	M0	340.724	126.029
126M	M1	343.317	128.621
325M	A0	325.112	325.112
325M	A1	325.112	325.112
325M	A2	327.391	327.391
325M	A3a	327.106	327.106
325M	A3b	326.821	326.821
325M	A3c	326.251	326.251
325M	A4	315.274	315.274

Pre-training. The larger grid uses AdamW with peak learning rate 4×10^{-4}, 170M warm-up tokens, cosine decay to 10% of the peak, sequence length 2048, batch size 1,048,576 tokens and bf16 computation. Its stored stream contains 8,406,403,457 tokens, with 63.05% FineWeb-Edu, 13.09% FineMath and 23.87% synthetic tokens. The smaller grid uses peak learning rate 6×10^{-4}, batch size 524,288 tokens, a 2% warm-up (94 iterations) and 4,730 training iterations. Its stream contains 2,487,505,384 tokens, with proportions 62.31%, 13.27% and 24.42%, respectively. Data-stream sizes and actually consumed training tokens are separately recorded in the metadata and run logs. Additional base seeds change model initialisation while sharing the pre-tokenised stream and its order. The synthetic generator samples both trajectory styles on style-enabled tasks; the style interventions therefore change SFT exposure, not whether the model has ever encountered the style.

SFT. The common 40k-example set uses task probabilities 0.4/0.2/0.2/0.2 for VARTRACK/SORT/LOOKUP/ARITH. STORY and BRTRACK each add 20k task-specific examples to this common set, giving 60k examples before length filtering. Training uses two epochs, peak learning rate 5×10^{-5}, 3% warm-up followed by cosine decay to 10%, batch size 64, and response-token loss. The implementation drops prompt-plus-response sequences exceeding 2048 tokens; available teacher and task-specific SFT logs report zero dropped examples. The reported 40k/60k counts are pre-filter counts. E4 changes the style label and response while preserving the underlying prompt and answer. Data manifests report counts, style fractions and SHA-256 hashes.

RL objective and implementation. Each rollout iteration samples 64 prompts and eight responses per prompt, with temperature 1.0 and at most 1536 new tokens. The default learning rate is 3×10^{-6}, with two optimiser minibatches per rollout iteration and held-out monitoring every 50 iterations. AdamW uses $\beta = (0.9, 0.999)$, $\epsilon = 10^{-8}$ and zero weight decay; gradients are clipped at norm 1. The clipped policy objective is token-normalised by default, using freshly recomputed old log probabilities from the training kernels. The entropy coefficient is 10^{-3} when mean rollout entropy is below 0.3; this bonus is active in 300/300 iterations of the main A0 and GDN 7:1 seed-0 runs and 298/300 for KDA. MoE also uses load-balancing and router regularisation terms. Thus "correctness-only" describes the task reward, not the entire optimisation objective.

Reward and style interventions. FORBID sets reward to zero whenever any reasoning line triggers the summary detector. State-style diagnostics use the stricter threshold of restatement rate > 0.5. The logged `forbid_zeroed_frac` is the number of otherwise-correct responses zeroed divided by *all* sampled responses, not by correct responses alone. A small value documents the prevalence of penalised outputs; it does not prove that the detector cannot be evaded. SSN uses population standard deviations within each (prompt, detected-style) subset, assigning zero advantage to singleton

or constant-reward subsets (Equation 2). Group retention still uses the original prompt-group reward variance, so zero-advantage style subsets can remain in token normalisation and entropy regularisation. The GDN 7:1 STORY SSN comparison changes generation microbatch size from 512 to 256; the core optimisation settings remain the same. The no-filter flag also explicitly drops zero-advantage samples; it is not an experiment that keeps all-equal groups in the loss denominator.

Prefix interventions and profiling. Prefix interventions recompute every cache from the edited token sequence, with positions assigned to that sequence; stripping therefore changes length and token positions as well as information. Filler preserves the replaced numeric characters' length, not necessarily their token count. Each condition receives up to 1536 new continuation tokens, in addition to the supplied prefix. The oracle replaces the last prefix summary with the true state after the first h source-program statements, where h is the retained prefix's reasoning-line count; this assumes line alignment, which was not independently checked. Profiling performs a warm-up and CUDA synchronisation around timed generation. Batch 1 uses 16 problems with one repeat, while batches 32 and 256 use 256 problems with three repeats. Each timed generation call includes prefill, cache allocation, CUDA-graph capture and its warm-up where used, autoregressive generation, and conversion of outputs to CPU lists. Prompt tokenisation and answer scoring are outside the timed region. Finished sequences remain in the microbatch until all sequences finish or the generation budget is reached. Caches and graphs are rebuilt per microbatch; this measures the supplied generation routine, not steady-state decode latency. The cost/accuracy plot combines these timings with accuracy from a separate evaluation grid, rather than measuring time per solved problem.

For BRTRACK at difficulty 20, original-completion versus reconstructed-full accuracy is 0.767/0.667 (full attention), 0.650/0.563 (GDN 7:1) and 0.896/0.771 (KDA 3:1). Prefix re-encoding and cache reconstruction can change the numerical generation path even without an intentional content edit; their individual effects were not isolated. Intervention differences are therefore measured against the reconstructed-full control.

Software snapshot. The project environment inspected during release preparation contains Python 3.12.14, PyTorch 2.13.0 (CUDA 13.0), FLA/fla-core 0.5.2, FlashAttention 2.8.3.post1, Triton 3.7.1, Transformers 5.16.1, Tokenizers 0.23.1 and NumPy 2.3.5. This is a packaging-time snapshot, not an immutable environment lock captured at every training run. Attention evaluation uses static KV caches and CUDA-graph decoding; recurrent kernels are provided by FLA.

B CRITERIA, PREDICTIONS AND OUTCOMES

The project was not publicly preregistered. An internal plan dated 12 September 2026 recorded the E1 non-inferiority margin, cost threshold and follow-up tests before the additional FORBID RL-seed results. The plan reused standard-RL runs and the first FORBID run that already existed. Results were later appended to the same notes, so file timestamps cannot verify an immutable registration history. We treat these as recorded predictions, not independently verified prospective commitments.

The E1 rule also required the difficulty-≥ 28 subgroup to pass separately. None of the four primary bases passed the -1 pp non-inferiority criterion on the full grid or this subgroup. The planned test therefore did not establish a lossless cost reduction. Later analyses describe the trade-offs without changing that outcome. Table 6 gives the criteria and results, including the failure of sequence-level loss to prevent switching. The mutable notes cannot certify the timing of every later prediction.

Table 6: Recorded criteria and predictions, with outcomes. The internal plan covers follow-up work after initial experiments (Appendix B).

criterion	recorded criterion/prediction	outcome
E1 non-inferiority of the terse path	one-sided 95% lower bound > -1 pp	Not met: KDA 3:1 -2.36 pp; GDN 7:1 -7.01 pp; controls A0/A3b $-1.39/-2.99$ pp. These are the 5th-percentile lower bounds, not the lower endpoints of two-sided CIs. Failure is not reinterpreted as equivalence.
E1 cost criterion	$\geq$ 20% fewer s/problem at batch 256	met: -65% (GDN 7:1), -68% (KDA 3:1)
E3 mechanism statistic	early retained-group style advantage agrees with subsequent style direction	Partial descriptive agreement; sequence-level loss switches only in the longer run. Positive advantages also occur without switching; no universal prediction rule is established.
E5 falsifiable prediction	sequence-level loss prevents or substantially delays the switch	No switch within 80 iterations; switching is observed by iteration 150, as in the standard run. Neither prevention nor a relative delay is established.
FORBID output audit	penalised correct responses as a fraction of all rollouts $< 5\%$	Low observed prevalence; this statistic does not certify detector completeness or exclude evasion.
E6 (SSN) prediction	stratified normalisation prevents the switch; accuracy near FORBID	**met** (9/9 runs across 325M, 126M and STORY; configuration-level mean differences within 1 pp of FORBID)
Other-base comparison	higher SFT accuracy is associated with terse policies; low accuracy predicts switching	Higher-accuracy bases finish terse in 12/12 standard-RL runs across three bands. The low-accuracy GDN 7:1 seed-2 base does not switch, despite later positive advantage windows: SFT accuracy and advantage sign are each insufficient alone.

C ADDITIONAL RESULTS

Table 7: E4: SFT style fraction $\rightarrow$ RL outcome (325M, held-out VARTRACK; final greedy style; paired difference versus standard RL from the 50% prior). Entries are means over n independent alternative runs ($n = 1$ unless shown), compared with the three standard runs. The 50% row is a separately trained control. At 100%, mean SORT accuracy is .470 / .477 for GDN 7:1 / KDA 3:1 (standard: .831 / .829).

state-style fraction in SFT	GDN 7:1	KDA 3:1	full attn
0%	.594 (terse, -4.0 pp, n=3)	.856 (terse, $+0.1$ pp)	.825 (terse, -0.6 pp)
25%	.633 (state, -0.1 pp)	–	.869 (terse, $+3.8$ pp)
50%	.656 (state, $+2.2$ pp)	–	.848 (terse, $+1.7$ pp)
75%	.650 (state, $+1.6$ pp)	–	.808 (state, -2.2 pp)
100%	.648 (state, $+1.4$ pp, n=2)	.863 (state, $+0.7$ pp)	.827 (state, -0.3 pp)

Table 8: Difficulty bands (325M, held-out VARTRACK 16–40; two RL seeds per cell for 24–40; one per cell for 40–64). No base switches on the 24–40 band. The 40–64 band on the seed-1 bases is evaluated on 16–64 statements.

band	base	standard acc.	FORBID acc.	style	FORBID − standard (pp)
24–40	GDN 7:1 s0	.600/.571	.579/.544	terse	-2.4 $[-4.9, -0.1]$
24–40	KDA 3:1 s0	.863/.879	.833/.831	terse	-3.9 $[-6.0, -1.7]$
24–40	GDN 7:1 s1	.960/.977	.971/.975	terse	$+0.4$ $[-0.7, +1.6]$
24–40	KDA 3:1 s1	.948/.971	.977/.967	terse	$+1.3$ $[+0.2, +2.4]$
40–64	GDN 7:1 s1	.953	.941	terse	-1.3 $[-3.3, +0.6]$
40–64	KDA 3:1 s1	.934	.941	terse	$+0.6$ $[-1.3, +2.5]$
24–40-band vs default-band standard RL (same held-out set): GDN 7:1 s0 -4.9 $[-7.7, -2.1]$; KDA 3:1 s0 $+2.3$ $[-0.1, +4.7]$					

Table 9: Style-stratified normalisation (SSN, E6) and STORY results (325M unless stated, held-out).

comparison	base	reference acc.	SSN acc.	paired diff. (pp)
SSN vs standard (VARTRACK)	GDN 7:1	.625/.644/.633	.588/.592/.590	-4.4 $[-6.5, -2.4]$
SSN vs FORBID (VARTRACK)	GDN 7:1	.579/.575/.592	.588/.592/.590	$+0.8$ $[-1.1, +2.6]$
SSN vs standard (VARTRACK)	KDA 3:1	.844/.852/.871	.844/.848/.829	-1.5 $[-3.6, +0.6]$
SSN vs FORBID (VARTRACK)	KDA 3:1	.848/.844/.854	.844/.848/.829	-0.8 $[-2.5, +0.9]$
SSN vs standard (126M, VARTRACK)	GDN 7:1	.838	.696	-14.2 $[-18.5, -9.8]$
SSN vs FORBID (126M, VARTRACK)	GDN 7:1	.688	.696	$+0.8$ $[-2.1, +3.8]$
SSN vs standard (STORY)	GDN 7:1	.790	.769	-2.1 $[-6.0, +1.9]$
SSN vs standard (STORY)	GDN 3:1	.883	.925	$+4.2$ $[+1.0, +7.3]$

Release test (E7), one run per architecture. GDN 7:1: FORBID checkpoint 0.579 → released 0.615 ($+3.5$ pp $[+0.2, +6.9]$); KDA 3:1: 0.848 → 0.865 ($+1.7$ pp $[-1.2, +4.6]$). Over 150 iterations (76,800 rollouts), the recorded state-style counts are 37 for GDN 7:1 and 0 for KDA 3:1. Neither greedy policy switches to summaries.

Style probe at the RL start. On VARTRACK, we sample eight rollouts per prompt from each SFT model on the training band. The two style accuracies pool all rollouts; parenthetical differences average the within-prompt state-minus-terse accuracy difference over prompts where both styles occur: seed-0 GDN 7:1 state 0.536 vs terse 0.519 (paired $+0.045$); seed-1 GDN 7:1 0.839 vs 0.851 (-0.004); seed-0 KDA 3:1 0.850 vs 0.768 ($+0.069$); seed-1 KDA 3:1 0.974 vs 0.935 ($+0.030$). On SORT the state style is 8–26 pp worse for every base, consistent with the observed disadvantage of the 100% prior on SORT (Table 7).

126M grid. Under the same recipe at 126M, GDN 7:1, KDA 3:1, GLA 3:1 and MoE+GDN 3:1 switch, while full attention, SWA-512, pure GDN, GDN 3:1, GDN 1:1 and attention-MoE stay terse (3 RL seeds each).

Probe limitation. The probe is exploratory. On STORY at the 325M scale, both tested bases switch despite negative initial paired state-style accuracy differences (approximately -2.6 pp each), illustrating a limit of the marginal-accuracy diagnostic.

D EVALUATION GRIDS AND STATISTICAL UNCERTAINTY

Evaluation design. Final paired comparisons use evaluation seed 777. The VARTRACK, SORT and STORY grids use difficulties 16, 20, 24, 28, 32 and 40, with 80 problems per difficulty (480 total per task). The BRTRACK grid uses 8, 12, 16, 20, 24 and 28. The 40–64 training-band experiment evaluates VARTRACK at 16, 20, 24, 28, 32, 40, 48 and 64 (640 problems). These final comparisons are distinct from the smaller in-training evaluation grid. Pairing uses matching evaluation seeds, task/difficulty order and item positions in the saved records; records do not contain hashes of the full prompts. This protocol assumes the shared deterministic generator and configuration used for the paired evaluations.

Primary intervals. The original analysis pools paired correctness differences across items and RL-seed pairs ($3 \times 480 = 1440$ pairs in the primary E1 cells) and takes the 2.5th and 97.5th percentiles of 4,000 bootstrap means. The non-inferiority test instead uses the 5th percentile as a one-sided 95% lower bound and compares it with -1 pp. The analysis uses NumPy's random generator initialised with seed 0; the supplied aggregate freezes the reported draws. Figure 1's difficulty-specific error bars use 1.96 standard errors rather than percentile bootstrap intervals. All intervals are descriptive, without adjustment for multiple exploratory comparisons.

Repeated items and training seeds. Pooling treats repeated evaluations of an item across RL seeds as separate observations. It therefore does not capture all dependence from shared prompts or trained policies, and should not be interpreted as uncertainty over independently pretrained architectures. Some recipe ablations have one alternative checkpoint paired with three standard checkpoints; this is not three independently trained alternatives. We report seed-level accuracies to expose that distinction.

As a sensitivity analysis during manuscript revision, we resample each shared item jointly across all available seed pairs (10,000 draws, seed 20260923), conditional on those trained policies. We also compute a two-sided Student-t interval from the paired RL-seed means, conditional on the fixed item set. With only two or three seeds, the latter intervals are imprecise; neither procedure estimates both sources of uncertainty simultaneously.

Table 10: Sensitivity of selected FORBID − standard comparisons to uncertainty treatment (pp). The shared-item bootstrap retains the same item across seeds. Seed-t intervals use the number of RL-seed pairs as the sample size.

task	base	mean	shared-item 95% CI	seed-t 95% CI
VARTRACK	GDN 7:1	-5.2	$[-7.7, -2.8]$	$[-8.8, -1.6]$
VARTRACK	KDA 3:1	-0.7	$[-3.2, +1.9]$	$[-3.3, +1.9]$
STORY	GDN 7:1	-2.6	$[-5.6, +0.6]$	$[-6.5, +1.3]$
STORY	GDN 3:1	$+3.9$	$[+1.4, +6.5]$	$[-2.8, +10.5]$

The main GDN 7:1 deficit remains below zero under both sensitivity analyses, while KDA 3:1 remains inconclusive. The STORY effects are less certain at the seed level; the positive GDN 3:1 point estimate should not be read as a precise population effect. For the primary three-seed comparisons, the correct seed-t half-widths are 3.62 pp (GDN 7:1) and 2.60 pp (KDA 3:1). Prefix-intervention files contain aggregate continuation accuracies rather than paired item-level outcomes, so no paired intervals are inferred for Table 3.

E Data and analysis provenance

Pre-training stream order. The 126M experiments used an earlier interleaving implementation that weighted source selection by remaining token budget. Because source documents have different mean lengths, this depleted long-document sources earlier and produced an unintended synthetic-heavy tail. The 325M stream used the corrected rule, which weights remaining token budgets by inverse mean document length to balance source depletion. The released builders contain that correction; they are not a byte-exact reconstruction of the earlier 126M stream. Both model scales use both trajectory styles in synthetic pre-training. We therefore interpret cross-scale comparisons jointly with their different data histories.

Records underlying the analyses. Final paired evaluations are separate from the in-training monitoring grid. Original per-item results and aggregate outputs are preserved in the supplement. The shared-item bootstrap and seed-level uncertainty checks in Appendix D were added during manuscript revision and are labelled separately from the original pooled analysis. Figures distinguish short diagnostic runs from longer training runs, and retain gaps for missing conditional statistics. The internal planning and errata records are included as historical documents; the methods and definitions in this manuscript describe the reported experiments.

Reconstruction limits. The saved configurations, logs, tokenizer and SFT hash manifest identify the available inputs and outputs. Training streams, complete SFT datasets and checkpoints are absent from this archive. The lookup generator also converts a Python set to a list, so exact regeneration of lookup-containing data requires the original hash-order environment as well as the random seed. This does not affect the four tasks used in the reported paired comparisons (VARTRACK, SORT, STORY, BRTRACK), whose released generators do not depend on that operation. A packaging-time environment snapshot and collection-time file hashes do not establish an immutable training environment or preregistration history.

F RUN AND ARTEFACT INVENTORY

Primary comparisons. At the larger scale, seven architectures have three standard-RL runs each. E1 reports three matched standard/FORBID RL-seed pairs for A0, A3a, A3b and A4; A1, A2 and A3c have one additional FORBID run each, not three. For A3a and A4, each of the two additional base initialisations has two standard and two FORBID RL runs. At the smaller scale, ten architectures have three standard runs each; the reported FORBID comparisons cover three seeds each for A3a, A4 and A5.

Interventions. E3 contains 80-iteration component ablations; longer E5 runs test sequence-level loss and the length-penalty baselines. E4 varies five SFT priors, with one to three RL seeds per reported setting. E6 includes six larger-scale VARTRACK runs, one smaller-scale run, and two STORY runs. E7 has one release run per hybrid. The 600-iteration comparison has one matched pair. BRTRACK uses A0/A3a/A4 with two seed pairs; STORY uses three pairs for A3a and two for A3b. The 24–40 band has four bases, each with two seed pairs; the 40–64 band has two base-seed-1 models with one pair each. Table 3 displays six prefix-intervention cells; the supplement also includes the additional KDA 32-statement cell. Six policies are profiled for inference cost.

Release contents and scope. The anonymous supplement separates frozen experimental records from revision-time analyses. It includes run arguments, model configurations, training logs, saved evaluation summaries, available per-item held-out records, timing/prefix outputs, the tokenizer and data manifests. Source code and launch commands describe the original experiments; release scripts recompute paired statistics and regenerate figures without retraining. Checkpoints and pre-tokenised corpus streams are excluded. Hash manifests identify the released files and recorded SFT datasets; they provide file identity, not a guarantee of independently reproduced training or a tamper-proof experiment history.