TRAINING WITH RECOVERABLE CONTEXT IMPROVES AGENTS BEYOND RECALL AT INFERENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Context management shapes not only what an agent can access at inference time but also what it can learn during reinforcement learning. When truncation causes all rollouts in a group to receive the same reward, group-relative advantage estimation removes that group's learning signal. We ask whether making truncated observations recoverable during training can improve the learned policy beyond the immediate benefit of recall at inference. Our method stores oversized observations outside the active context, replaces them with compact references, and allows the policy to retrieve them under explicit budgets, while treating recalled text strictly as observation rather than a prediction target. Across four model scales, recall-enabled training improves mathematical accuracy even when recall is disabled at evaluation, indicating that recoverable context changes what the policy learns, not just what it can access. On long-context tasks, it also substantially increases answer coverage. Controlled experiments rule out unprompted recall behavior and generic long-context reasoning as sufficient explanations. These findings establish context management as a training intervention, not merely an inference-time design choice, and highlight the importance of separating its learning effects from its retrieval benefits.

1 INTRODUCTION

Context management Liu et al. (2026) is part of the learning environment for tool-using agents. A policy issues a tool call, observes its result, and uses that evidence to select subsequent actions. When an observation is truncated to satisfy a context budget, however, the policy must act on incomplete information. Thus, agentic reinforcement learning (RL) Zhang et al. (2026) depends not only on which tasks are sampled but also on which observations remain accessible during rollouts.

Prior work has improved tool-integrated training and learned memory management (Yu et al., 2026b), but a complementary question remains: *what happens to learning when a potentially useful observation is discarded during training?* In group-relative policy optimization (Shao et al., 2024), rewards are compared across multiple rollouts of the same task. If truncation removes evidence required for success, a task may produce an all-failure group with zero centered advantages, consuming compute while contributing no reward-based policy-surrogate signal. This concerns the reward-driven term rather than the optimizer or regularization as a whole.

This motivates our central question:

> *Can recoverable tool observations during RL improve the learned policy beyond the direct inference-time benefit of recall?*

We study this question by integrating address-based observation recovery into Agent0's rollout manager (Xia et al., 2026; Dang et al., 2026). Before truncating an oversized observation, the manager stores its full content externally and replaces it with a compact reference. The policy can later retrieve the content through an explicit recall action, subject to budgets on recalled text. Retrieved text is treated strictly as an environment observation and excluded from policy prediction targets, so the intervention changes observation availability without changing the optimizer or model architecture.

We compare four training conditions under three inference configurations across four backbones from two model families. Recall-enabled training improves mathematical accuracy in 10 of 12

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.

matched comparisons. The portion of this gain that persists when recall is disabled at evaluation appears on both Qwen3 and Gemma models, while an untrained-policy control shows that a substantial portion of the long-context benefit is an inference-time effect available without training. These results indicate distinct training-associated and inference-time benefits.

Our contributions are threefold: (1) we identify how observation loss can eliminate group-relative reward signal; (2) we introduce observation recovery that preserves recalled content as environment observation rather than generated supervision; and (3) we develop an evaluation design that separates training effects from inference-time recall and protocol disclosure. More broadly, our results suggest that context management is not only an inference-time systems decision but also a component of the training environment that can shape what a policy learns.

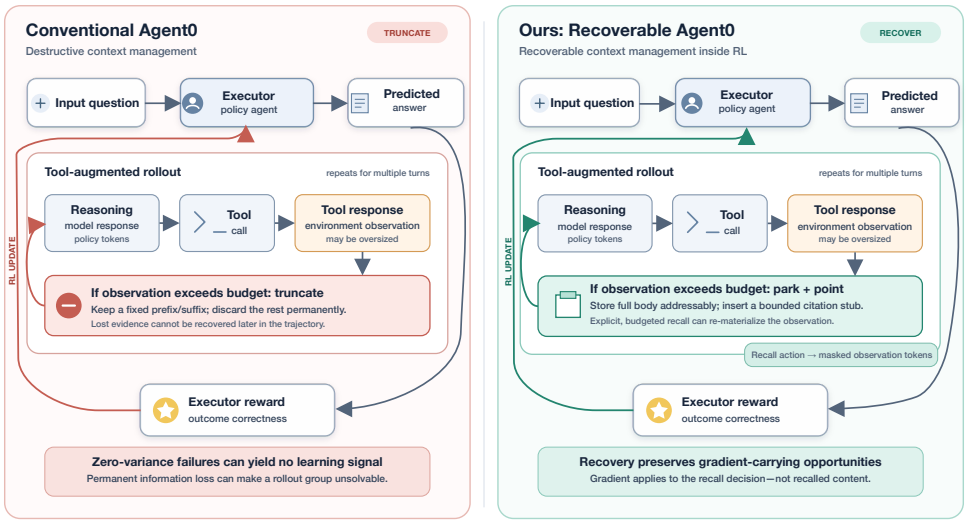

Figure 1: Our recoverable context management method vs truncation method.

2 Background

2.1 Actions, observations, and context limits

For a task prompt x, a policy π_θ interacts with an environment through alternating model-generated *actions* and environment-provided *observations*, such as tool outputs. The resulting sequence forms a *trajectory* until termination.

Two limits determine what information remains available: an *observation budget* limits the text retained from an individual tool response, while the *context limit* bounds the total input visible to the model. Thus, a single tool response may exceed the observation budget even when the overall trajectory fits within the context limit.

In the baseline, oversized observations are truncated destructively: discarded content cannot be accessed later, potentially removing evidence needed for reward. We instead use *recoverable observations*: the full content is stored outside the active context and replaced by a compact reference, which the policy can later retrieve through a bounded recall operation. Recoverability preserves access to discarded information without removing the context bound or guaranteeing task solvability.

Thus, our approach makes information preservation part of the interaction environment while keeping the model's active context bounded.

2.2 GROUP-RELATIVE REWARDS

Our training procedure uses a group-relative policy objective. For each task, $n \geq 2$ trajectories are sampled with terminal rewards $r_1, \ldots, r_n \in \mathbb{R}$. Let

$$\bar{r} = \frac{1}{n}\sum_{i=1}^{n} r_i, \qquad \sigma_r = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(r_i - \bar{r})^2}, \tag{1}$$

denote the within-group reward mean and standard deviation. The normalized advantage is

$$A_i = \frac{r_i - \bar{r}}{\sigma_r + \delta}, \qquad \delta > 0, \tag{2}$$

following GRPO (Shao et al., 2024), which replaces PPO's learned value function (Schulman et al., 2017) with within-group normalization. Thus, learning depends on both absolute rewards and their variation within each sampled group.

Our experiments use Agent0's Ambiguity-Dynamic Policy Optimization (ADPO) (Xia et al., 2026), which further rescales advantages and adapts likelihood-ratio clipping based on agreement among sampled answers. Its *self-consistency score* is the relative frequency of the most common answer category and measures sample agreement, not ground-truth accuracy. Our observation-management mechanism leaves ADPO and the underlying optimization algorithm unchanged.

2.3 WHY DISCARDED OBSERVATIONS MATTER FOR LEARNING

Group-relative objectives exhibit a simple failure mode relevant to tool-using agents. If all trajectories in a sampled group receive the same reward,

$$r_1 = \cdots = r_n = c, \tag{3}$$

then

$$\bar{r} = c, \qquad \sigma_r = 0, \qquad A_1 = \cdots = A_n = 0. \tag{4}$$

Thus, the reward-dependent policy objective receives no group-relative contribution from that group.

Observation truncation can cause this when discarded text contains evidence required for success. If all rollouts lose the same evidence and receive the same failure reward, the task consumes rollout computation without producing a reward-based learning signal. However, equal rewards do not imply truncation caused the failure; they may also result from uniformly successful trajectories or other causes.

Here, *reward-based training signal* refers specifically to the reward-dependent policy objective, not the complete parameter update, which may also include regularization or contributions from other groups.

Our method therefore preserves potentially reward-relevant observations while leaving the underlying RL objective and optimizer unchanged.

3 RELATED WORK

Self-evolving agents trained from zero data. R-Zero (Huang et al., 2026) improves reasoning from zero external data via a co-evolutionary loop between a Challenger (problems at the Solver's capability edge) and a Solver trained under majority-vote pseudo-labels. It is purely text-based and non-agentic—no tool calls or environment observations—so the context-overflow failure mode studied here is architecturally out of scope. We evaluate an R-Zero checkpoint per backbone as an external reference, not a controlled contrast. Agent0 (Xia et al., 2026) extends the design to tool use, pairing a GRPO-trained Curriculum Agent with an ADPO-trained Executor. Agent0 is our experimental substrate; its fixed per-observation truncation, hard context ceiling, and loss-masking of tool outputs leave no recovery path and never treat truncation as a variable to ablate. **Context management for long-horizon agents.** ARC (Dang et al., 2026) replaces irreversible compaction losses with a store-and-cite construction: observations are written verbatim to an append-only, ID-addressable log, oversized ones appear as bounded citations, and an explicit recall action re-materializes the

original. Like the broader long-context literature, its evaluation is inference-only on a frozen policy, with no analogue of an advantage estimator. We are unaware of prior work that varies a context-management policy as a controlled training-time condition. **Degenerate advantage groups, and two opposite design choices.** Systems already respond to zero-variance groups: DAPO (Yu et al., 2026a) oversamples and filters uniform groups to keep gradient-carrying groups stable, while curriculum schemes (including ADPO) steer data toward high-variance bands. Both treat degeneracy as intrinsic to the task–policy pair; we argue one source is manufactured by the rollout substrate, and repairing it recovers signal that filtering can only discard. Two further lines make the opposite architectural choices (treated fully in the Appendix). ACC (Su et al., 2026) unmasks tool responses for agent SFT—correct for single-shot SFT but problematic in group-relative RL—while AdaCoM (Yi et al., 2026) and ActiveContext (Liu et al., 2026) train a separate context manager against a frozen agent to avoid gradient interference. Our masking invariant is the design response to the latter objection, stated as a rationale rather than a demonstrated mechanism.

4 Recoverable Observations for Agentic RL

We propose a simple change to the interaction interface of a tool-using agent: observations that would otherwise be discarded are preserved outside the active context and can be explicitly recovered by the policy when needed. The method changes how information is exposed during a rollout, rather than modifying the model architecture or the underlying reinforcement-learning algorithm (Figure 1). The design follows two principles. First, context management should not make potentially useful evidence irreversibly unavailable. Second, recovered text remains an environmental observation and should not be treated as content generated by the policy. We realize these principles through a recoverable observation interface and an action-only learning objective.

4.1 Recoverable observation interface

Preservation. When a tool observation exceeds the active context budget, a destructive context manager shortens it and permanently discards the omitted content. Our method instead stores the original observation in a trajectory-local *archive* and replaces it with a compact *observation reference* containing its identifier and a bounded preview. Observations that fit remain unchanged. Thus, the archive preserves trajectory information while the active context contains only information currently exposed to the policy.

Recovery. The policy can retrieve an archived observation through an explicit *recall action* specifying its identifier. The environment resolves the identifier and returns the archived content without re-executing the original tool. Recall is therefore a policy decision rather than an automatic context-management operation, separating information preservation from information use. The returned content can also be bounded, so recoverability does not require restoring arbitrarily large observations to the active context.

Bounded active context. Let $B_{\mathrm{ret}} > 0$ denote an upper bound on the amount of text returned by a single recall, and let $B_{\mathrm{active}} > 0$ denote an upper bound on the total recalled content simultaneously retained in the active context. If the currently active recalled observations are $u_1, \ldots, u_q$, we require

$$\ell_{\mathrm{char}}(u_j) \le B_{\mathrm{ret}} \quad \text{for } j = 1, \ldots, q, \qquad \sum_{j=1}^{q} \ell_{\mathrm{char}}(u_j) \le B_{\mathrm{active}}, \tag{5}$$

where $\ell_{\mathrm{char}}(u)$ denotes the length of text u in characters.

When the active recalled content would exceed its budget, previously recalled content can be removed from the active context while its archived copy remains available. Preservation is therefore persistent over the trajectory even when visibility is temporary.

The return budget and the model's context limit are conceptually distinct. The former controls how archived observations are exposed through recall, whereas the latter constrains the complete input presented to the model. Our method requires only that the recall interface respect both constraints; their concrete values are implementation choices.

4.2 LEARNING FROM RECALL DECISIONS

Recall changes the policy's evidence, but the returned text is environment-generated. The objective should credit policy decisions, including recall, without treating this text as policy output.

For trajectory i, let $\mathcal{A}_i$ contain the policy-generated token positions, excluding tool observations, environment-inserted references, and recalled text. For each $t \in \mathcal{A}_i$, let y_{it} be the generated token and h_{it} its visible history.

The objective is evaluated only on $\mathcal{A}_i$. For a likelihood-ratio-based objective,

$$\rho_{it}(\theta) = \frac{\pi_\theta(y_{it} \mid h_{it})}{\pi_{\theta_{\text{old}}}(y_{it} \mid h_{it})}, \qquad t \in \mathcal{A}_i, \tag{6}$$

where θ_{old} generated the trajectory. Environment-provided text receives no likelihood ratio.

For the group-relative setting introduced in Section 2.2, a generic reward-dependent objective is

$$J_x(\theta) = \sum_{i=1}^{n} \sum_{t \in \mathcal{A}_i} w_{it}\, \mathcal{L}_{it}(\theta, A_i), \tag{7}$$

where $w_{it} \geq 0$ is an aggregation weight and $\mathcal{L}_{it}$ is the algorithm's per-token policy term. The method requires only that environment-generated text be excluded from prediction targets.

Recalled observations still affect learning through the history h_{it} of later actions and thus their decisions and gradients. Only direct credit assignment to the observation text is excluded. The environment preserves and returns evidence, while the policy learns whether and how to use it, allowing recall to participate in reinforcement learning without conflating policy behavior with environment content.

5 EXPERIMENTAL SETUP

5.1 IDENTIFICATION: THE TRAINING × INFERENCE × DISCLOSURE FACTORIAL

Four training conditions.

Per model we compare four methods: **Base** (Yang et al., 2025; Team et al., 2024) — the untrained model, no RL. The control that makes prompt-disclosure artifacts detectable. **R-Zero** (Huang et al., 2026) — zero-data self-evolution agent. An external reference for "does zero-data RL help at all on this method," not a controlled contrast: R-Zero's own training configuration is not matched to ours since we built it in on a different mechanism. **Agent0** (Xia et al., 2026) (truncate) — oversized observations are tail-truncated to a fixed token budget, permanently, exactly as in unmodified Agent0's executor. This is the matched baseline. **Ours** (recall/recall-silent as mechanism available, protocol not disclosed) — oversized observations are stored, cited, and recallable (§4).

5.2 THE MEMORY-STRESS TASK

To obtain a diagnostic solvable if and only if truncated content can be recovered, we construct a single-needle recall task. A dedicated tool returns a noise haystack whose length *exceeds* the 512-token observation budget, so the observation is always oversized and the substrate always fires: under **truncate** the overflow is discarded permanently, and under our method it is archived and replaced by a citation stub. A short random token—the needle—is planted in the *middle* of the haystack, placing it in the discarded region under **truncate** and outside the stub's bounded head/tail preview under our method, so it is obtainable only by an explicit recall against the archived observation. The tool serves the haystack exactly once per trajectory, keyed so each of a prompt's n parallel rollouts gets an independently fresh read, and returns an expired decoy on any subsequent request, closing the re-request loophole. By construction the task is solvable under **recall** and unsolvable under **truncate** regardless of training: a mechanism-existence check.

5.3 EVALUATION BENCHMARKS

LongBench-v2 (Bai et al., 2025): using 311 hard samples, 4-way multiple choice, 25.0% chance accuracy, served through the same tool loop and substrate used in training at max model length=8,192

Table 1: Training dynamics, four backbones × two methods (A0 = Agent0, Ours = recall). Mem-stress reward and val math reward report the "first | last" gives both endpoints over 100 optimizer steps. Em-dashes mark quantities absent from A0.

Metric	Q3-4B A0	Q3-4B Ours	Q3-8B A0	Q3-8B Ours	G3-4B-it A0	G3-4B-it Ours	G2-9B-it A0	G2-9B-it Ours
mem-stress reward	0.00|0.00	0.01|**0.45**	0.00|0.00	0.00|**0.72**	0.00|0.00	0.52|0.02	0.00|0.00	0.00|0.00
math val reward	0.56|0.62	0.57|**0.64**	0.54|0.62	0.47|**0.63**	0.65|0.69	0.67|**0.70**	0.25|0.48	0.14|**0.54**
max actor KL loss	0.007	**0.066**	0.008	**0.014**	0.084	**0.105**	0.139	**0.339**
Δ mean res-tokens	+107	**+282**	+59	**+285**	−170	**+404**	+43	**+217**
recall actions	—	14.45%	—	11.40%	—	13.82%	—	13.16%

and max tokens=2,048. **A seven-benchmark math tasks**: GSM8K (1,319 problems) (Cobbe et al., 2021), MATH-500 (500) (Hendrycks et al., 2021; Lightman et al., 2024), AIME24 (30), AIME25 (30), AMC-23 (40) (Li et al., 2024), Minerva (272) (Lewkowycz et al., 2022), and OlympiadBench (675) (He et al., 2024), totaling 2,866 problems, evaluated through the full agentic tool loop. We report the pooled micro-average, the unweighted macro-average, and a non-GSM8K pooled average, since GSM8K is 46% of the pool by problem count and near ceiling for these backbones, and can otherwise mask movement on harder benchmarks. Section 6.3 shows this is not a hypothetical concern: one baseline's entire pooled gain is a GSM8K effect that reverses sign on the remaining 1,547 problems.

6 RESULTS

6.1 RL TRAINING PHASE

Table 1 summarises the key metrics from the eight runs (full metric set in Appendix I, Table 15; per-model training curves in the same section).

Three signals hold in all four models. Our method's **KL divergence from the reference policy travels further** in every pair, by factors of 1.3–9.5× (Appendix Table 15) — consistent with exposure to a larger pool of gradient-carrying signal. Our method's **final held-out math validation reward is higher** in all four pairs (+0.006 to +0.060), measured on the same shared validation set so it is not mechanically inflated by the memory-stress subset (peak validation reward, a weaker signal we do not rest on, favors our method in only 3 of 4 pairs). **Mean response length grows more** in our method in all four pairs, and grows at all only in our method on Gemma-3-4B-it, where Agent0's responses *shrank* by 170 tokens — suggestive that a policy able to recover content produces longer trajectories, while one that cannot learns to stop trying.

How much signal each run recovered. Peak and final memory-stress reward tell different stories: three of four backbones reach a high *peak* (0.86, 0.72, 0.97) 15, so the substrate works on all three, but only two *end* training able to solve the subset — Gemma-3-4B-it climbs to 0.97 mid-training then collapses to 0.02 by the final checkpoint even as recall-action rate keeps rising (Appendix I), and Gemma-2-9B-it never scores above zero. Since we evaluate final checkpoints, the final value is the dose that should predict downstream gains (§6.3, §8). We observe that instruction-based models such as Gemma exhibit different response patterns, leading to behaviors that differ from those of Qwen models.

6.2 LONGBENCH-V2 TASKS

Table 2 reports abstention — the fraction of the 311 hard tasks yielding no extracted answer — pooled, and for the two extreme cells; the full 12-cell breakdown with per-cell *z/p* is in Appendix C, Table 8.

When applied during inference on a fixed-parameter model, this mechanism reduces the rate of context-based refusals by approximately two-thirds (based on aggregated data); refusal rates also decreased in 10 out of 12 case groups under the "recall-silent" mode. The two exceptions were due to floor effects at the "Agent0" and "Ours" checkpoints on the Qwen3-8B model, where the models exhibited zero refusals out of one item in the truncation mode.

Table 2: LongBench-v2 abstention (pooled and extreme cells), z/p compare **recall-silent** against **truncate**.

Model	Method	truncate	recall-silent	recall	Δ	z	p
Qwen3-8B	Base	**47.9% (149)**	**4.8% (15)**	6.8% (21)	-43.1 pp	-13.98	$< 10^{-40}$
Qwen3-4B	Ours	1.0% (3)	0.6% (2)	2.6% (8)	-0.3 pp	-0.45	0.65
Pooled ($n = 3{,}732$)		**10.7% (401)**	**3.6% (134)**	6.2% (231)	-7.2 pp	-12.10	$< 10^{-30}$

The extent of the impact is proportional to the severity of the underlying issue: in the case of the largest model, the **untrained Qwen3-8B**, the abstention rate was 47.9% under length constraints and 4.8% with the support mechanism; this occurred because the untrained 8B model, when faced with an 8,192-token limit, generated unstructured reasoning chains that caused memory overflows and premature termination before an answer could be produced. Both RL training methods largely resolved this error, and Qwen3-8B versions trained via the Agent0 method or our own method virtually ceased abstaining under length constraints—indicating that, for this model, the coverage benefits provided by the mechanism were achieved through training rather than serving as an additive factor. We attribute this to RL helping to standardize the format of the reasoning chain rather than shortening it; in fact, the average response length actually increased during training for both methods on this model (Table 1), so the reduction in abstention rates was not a result of length modification. Gemma-2-9B-it presents the opposite scenario: the model exhibited high abstention rates across all conditions (12–20%), and while the support mechanism helped reduce this rate, it was far from eliminating it entirely.

6.3 The Math evaluation

Table 3 reports the row effect on pooled accuracy over the seven-benchmark, 2,866-problem math suite: training with versus without the mechanism, at matched inference condition.

Table 3: Training effect on math downstream tasks pooled micro-accuracy (percentage points), Ours-trained vs Agent0-trained at matched inference condition. Non-GSM8K pooled deltas in parentheses.

Model	truncate	recall-silent (Ours)	recall (Ours)	mean
Qwen3-4B	$+1.66$ ($+1.63$)	$+2.23$ ($+3.70$)	$+1.93$ ($+2.13$)	$+1.94$
Qwen3-8B	$+1.58$ ($+0.53$)	$+5.18$ ($+2.51$)	$+2.05$ ($+1.50$)	$+2.93$
Gemma-3-4B-it	-0.06 (-0.20)	$+0.82$ ($+1.27$)	$+1.02$ ($+1.30$)	$+0.59$
Gemma-2-9B-it	-0.01 (-0.86)	$+0.45$ ($+1.26$)	$+1.02$ ($+0.53$)	$+0.49$
Pooled ($n = 11{,}464$)	$+0.79$, $p = 0.22$	$+2.17$, $p = 7.8 \times 10^{-4}$	$+1.50$, $p = 0.020$	$+1.49$

Ten out of the twelve comparisons yielded positive results, with no instances of significant performance degradation. The two cases that did not yield positive results (showing differences of -0.06 and -0.01 percentage points, respectively, out of 2,866 problems) corresponded to differences of 2 and 0 in the number of problems solved; thus, they are considered ties. A two-sided sign test yields $p = 0.002$ when treating these cases as ties. Broken down by benchmark (12 comparisons $\times$ 7 benchmarks $= 84$ cases; while not independent, these provide detailed insights into data characteristics), the results are distributed as follows: 51 improvements, 20 degradations, and 13 ties ($p = 3.0 \times 10^{-4}$). These changes are primarily concentrated in the GSM8K (11/12), MATH-500 (9/12), and OlympiadBench (9/12) benchmarks, while being virtually absent in the 30-problem AIME sets, where a change in a single problem can alter accuracy by 3.3 points.

This separation is related to dose. Ranking the backbone models by their average row effect gives: Qwen3-8B ($+2.93$), Qwen3-4B ($+1.94$), Gemma-3-4B-it ($+0.59$), and Gemma-2-9B-it ($+0.49$). Ranking them instead by the recovered "dead" signal, measured by the final memory pressure reward for the test group (Table 1), gives 0.719, 0.453, 0.016, and 0.000. The two rankings match exactly. In other words, the models that do not show the redundant **truncation** effect are also the models that finish training unable to solve the recall-dependent subset. We report it because the pattern can be tested. It also shows why saying that this mechanism works better on Qwen3 but Gemma-3-4B-it

Table 4: LongBench-v2 accuracy, mean ± SD over three training seeds (42, 82, 122). Δ is Ours vs Agent0 at matched inference condition.

Backbone	Inference	Agent0 (%)	Ours (%)	Δ
Qwen3-4B	truncate	22.20 ± 0.00	23.50 ± 0.30	$+1.30$
Qwen3-4B	recall-silent	24.57 ± 0.92	26.60 ± 1.61	$+2.03$
Qwen3-4B	recall	28.93 ± 1.70	29.70 ± 1.06	$+0.77$
Qwen3-8B	truncate	12.33 ± 0.81	$\mathbf{16.30} \pm 1.15$	$\mathbf{+3.97}$
Qwen3-8B	recall-silent	24.67 ± 1.02	24.13 ± 0.95	-0.54
Qwen3-8B	recall	24.43 ± 0.65	21.30 ± 2.34	-3.13

reached the *highest* peak memory pressure across all runs (0.969) before collapsing, yet its final result was zero.

Two comparisons at the matched **truncate** condition across all four training methods (Base, R-Zero, Agent0, Ours; full per-model, per-benchmark numbers in Appendix D, Table 9) are worth drawing out. R-Zero's pooled micro-average is 3.07 points above the untrained Base ($p = 2.6 \times 10^{-6}$), but its non-GSM8K pooled average is 0.31 points *below* it: R-Zero's entire measured gain in our harness is concentrated on the 1,319 GSM8K problems, which are 46% of the pool (most dramatically on Gemma-2-9B-it, where R-Zero lifts GSM8K from 69.1% to 87.7% while AMC-23 falls from 17.5% to 7.5% and OlympiadBench from 17.2% to 15.4%). Agent0 and Ours both gain on GSM8K *and* on the remaining 1,547 problems ($+2.34$ and $+2.61$ pp respectively over Base). Comparing Ours against R-Zero directly (Appendix D, Table 10), Ours leads on all four backbones at the matched **truncate** condition (pooled $+1.74$ pp, $p = 0.007$; non-GSM8K $+2.92$ pp, $p = 0.0008$) and by a wider margin under an oracle over the three inference conditions (pooled $+3.44$ pp, $p < 10^{-7}$).

7 Ablation

7.1 Seed robustness

To check whether the row effects are seed noise, we retrained both Qwen3 models on three seeds (42, 82, 122) — the two on which the load-bearing **truncate** effect is non-null — and re-evaluated the full LongBench-v2 and math tasks.

Regarding the math test (full details in Appendix F), our method outperforms Agent0 in all six comparison scenarios (across combinations of foundation models and inference conditions): $+1.97$, $+2.44$, and $+1.84$ for the Qwen3-4B model, and $+2.77$, $+7.27$, and $+1.67$ for the Qwen3-8B model, corresponding to the **truncate**, **recall-silent**, and **recall** modes, respectively. Each margin of difference is several times larger than the higher standard deviation of the two conditions, and a two-sided sign test across these six comparisons yields a p-value of 0.031. This constitutes the paper's strongest evidence supporting the claim regarding the training phase; this is most clearly demonstrated by the pivotal **truncate** comparison—a scenario where the mechanism is architecturally entirely absent during evaluation—which shows differences of $+1.97$ (standard deviations: 0.50, 0.12) and $+2.77$ (standard deviations: 1.39, 0.35). Both of these results fall entirely outside the range of noise attributable to the initialization value, whereas the corresponding effects observed in the main results in Table 3: $+1.66$ and $+1.58$.

On LongBench-v2 (Table 4). Ours leads at all three conditions on Qwen3-4B ($+0.77$ to $+2.03$ pp) and decisively at **truncate** on Qwen3-8B ($+3.97$ pp), but Agent0 at **recall-silent** (-0.54) and **recall** (-3.13) on that backbone, the latter by more than its own seed SD. The one contrast positive in all four (backbone × benchmark) combinations is **truncate**: $+1.30$/$+3.97$ on LongBench-v2, $+1.97$/$+2.77$ on math.

7.2 Sensitivity to context-length and response-length budgets

The two recall-specific budgets: a 16,000-character trajectory-level recall budget and an 8,000-character per-recall cap, are stated in absolute characters rather than as a fraction of the context

window, so it is not obvious in advance that the inference-condition ordering reported in §6.3 is robust to a different length-budget choice rather than an artifact of the one we picked. We test this on Qwen3-4B by training five Ours checkpoints that cross the training-time context ceiling max model length $\in \{8{,}192; 10{,}240\}$ with max-response length $\in \{2{,}048; 3{,}072; 4{,}096\}$ tokens. Each method is evaluated on the full math tasks under all three inference conditions and, for three of the five, at more than one eval-time max-response length, for 27 (method $\times$ eval-budget $\times$ inference-condition) cells; all three columns are the *same* Ours-trained checkpoint. (Appendix G, Table 13)

Repeating the design on LongBench-v2 ($N = 311$ hard tasks) with the same methods across every eval-time budget and all three inference conditions (36 cells; full results in Appendix H, Table 14) gives no single ordering that survives every cell: across the 12 (checkpoint $\times$ eval-budget) cells, disclosed **recall** is best or tied-best in 8 and worst in only 2, **truncate** is worst in 8 and best in only 2, and **recall-silent** is best or tied-best in 4, disclosing the recall protocol tends to *help*.

8 Discussion and Conclusion

Truncated context management is usually treated as an inference-time engineering detail, but our results suggest it is also a property of the training distribution. At inference, without retraining, our mechanism reduces context-driven abstention by roughly two thirds while leaving conditional answer quality largely unchanged; this effect appears on both trained and untrained backbones. At matched inference conditions, training with our method improves math accuracy in 10 of 12 comparisons across four backbones.

By construction, roughly 14% of our training mixture is **structurally unsolvable under destructive truncation**: facts discarded by truncation cannot be recovered through correct reasoning. Under group-relative advantage estimation, these tasks form zero-variance reward groups that consume rollout budget but provide neither positive nor negative gradient. Our method can convert these dead groups into solvable, gradient-carrying ones, changing the fraction of the training distribution that can contribute a learning signal.

Across four models, §6.3 agrees with this prediction, while the two model with no residual effect are those whose final runs failed to solve the recall-dependent subset. Gemma-3-4B-it provides a useful stress case: despite reaching the highest memory-stress peak (0.969), it ends with a null effect, consistent with final rather than peak memory stress determining the available signal.

Three qualifications matter: the memory-stress subset has zero Agent0 reward across all models, so we rely on held-out math validation reward instead, where our method finishes higher in all 4 pairs. RL is extremely sensitive to which parts of the training data distribution generate gradients, and context management can determine that aspect.

9 Limitations

There is no theoretical guarantee distinguishing between recall and recall-silent format. Under **recall-silent** the mechanism is available but the protocol is never disclosed, and this is the best-performing condition on the math tasks in all 27 cells of §7.2.

There is no larger-observation-budget control. Our method gives the policy access to text that truncation would discard. We maybe ask whether simply raising the 512-token observation budget would capture the same gain. We do not run that setting, so we cannot separate "recoverability helps" from "more observation text helps".

The Gemma-2-9B-it null may be a harness effect rather than a backbone property. This run issues recall actions on 13.16% of rollout actions — comparable to the three based models that do solve the task — and yet scores exactly 0.000 on the memory-stress subset at every logged step. It also abstains heavily in every LongBench-v2 condition (12–20%) and sits below chance in all nine of its cells.

REPRODUCIBILITY STATEMENT

Every hyperparameter used in the reported runs is given in Appendix B; the paired-training protocol, the four training conditions and the three inference conditions are specified in §5; the memory-stress task construction is specified in §5.2; and the statistical procedures, including how multiplicity is handled, are specified in §5.3. All 40 LongBench-v2 cells and all 32 math-suite cells underlying the main grid are reported per-cell in Appendices C and D, and the full length-budget sweep is reported per-cell in Appendices G and H, so every aggregate in the main text can be recomputed from the appendices. Training seeds are 42, 82 and 122 for the sweep of §7.1.

AI USE STATEMENT

We used generative AI assistants as supporting tools throughout this work. They helped with code navigation, implementation and debugging, experiment documentation, and language editing (including drafting parts of the manuscript). The research questions, experimental design and protocols, evaluation criteria, analyses, and all substantive claims were solely determined by the authors. Every AI-generated piece of code or text was carefully reviewed and revised by the authors as needed. All reported measurements come directly from the experimental artifacts and were verified against the original logs and result files. The authors accept full responsibility for the content of the paper and its accompanying artifacts.

ETHICS STATEMENT

This work presents a technical study of context management as a training-time intervention in reinforcement learning for language models. It does not involve human subjects, the collection or release of new datasets containing personal or sensitive information, or applications with direct potential for harm. All experiments were conducted on standard public mathematical reasoning and long-context benchmarks using publicly available model checkpoints. We identify no privacy, security, fairness/discrimination, dual-use, or legal-compliance concerns arising from the methods or results. The research focuses on understanding how truncation and recoverable observations affect learning dynamics under group-relative advantage estimation; it does not introduce new capabilities that meaningfully alter the risk profile of existing language models. The authors declare no conflicts of interest or external sponsorship that could influence the reported findings. All experimental measurements were obtained from logged artifacts and verified by the authors, who take full responsibility for the integrity of the results.

REFERENCES

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, et al. Longbench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3639–3664, 2025.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Thang Dang, Yuma Ichikawa, Sakina Fatima, and Koichi Shirahata. Addressable recall compaction for long context-window control in ai agents. *arXiv preprint arXiv:2607.25066*, 2026.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, 2024.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Chengsong Huang, Wenhao Yu, Xiaoyang Wang, Hongming Zhang, Zongxia Li, Ruosen Li, Jiaxin Huang, Haitao Mi, and Dong Yu. R-zero: Self-evolving reasoning llm from zero data. In *International Conference on Learning Representations*, volume 2026, pp. 130770–130790, 2026.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857, 2022.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, et al. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*, 13(9):9, 2024.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024.

Shukai Liu, Bo Jiang, Jian Yang, Yizhi Li, Jinyang Guo, Xianglong Liu, and Bryan Dai. Context as a tool: Context management for long-horizon swe-agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 20604–20617, 2026.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025.

Qisheng Su, Zhen Fang, Shiting Huang, Yu Zeng, Yiming Zhao, Kou Shi, Ziao Zhang, Lin Chen, Zehui Chen, Lijun Wu, and Feng Zhao. Acc: Compiling agent trajectories for long-context training. *arXiv preprint arXiv:2605.21850*, 2026. URL https://arxiv.org/abs/2605.21850.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Peng Xia, Kaide Zeng, Jiaqi Liu, Can Qin, Fang Wu, Yiyang Zhou, Caiming Xiong, and Huaxiu Yao. Agent0: Unleashing self-evolving agents from zero data via tool-integrated reasoning, 2025. *URL https://arxiv. org/abs/2511.16043*, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Lu Yi, Runlin Lei, Liuyi Yao, Yuexiang Xie, Yuyang Li, Wenhao Zhang, Zhewei Wei, Yaliang Li, and Jian-Yun Nie. Learning agent-compatible context management for long-horizon tasks. *arXiv preprint arXiv:2605.30785*, 2026. URL https://arxiv.org/abs/2605.30785.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026a.

Yi Yu, Liuyi Yao, Yuexiang Xie, Qingquan Tan, Jiaqi Feng, Yaliang Li, and Libing Wu. Agentic memory: Learning unified long-term and short-term memory management for large language model agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21457–21483, 2026b.

Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, Zaibin Zhang, Zelin Tan, Heng Zhou, Zhong-Zhi Li, Xiangyuan Xue, Yijiang Li, Yifan Zhou, Yang Chen, Chen Zhang, Yutao Fan, Zihu Wang, Songtao Huang, Francisco Piedrahita Velez, Yue Liao, Hongru WANG, Mengyue Yang, Heng Ji, Jun Wang, Shuicheng YAN, Philip Torr, and LEI BAI. The landscape of agentic reinforcement learning for LLMs: A survey. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=RY19y2RI1O. Survey Certification.

A Mathematical Details and Assumptions

This appendix makes explicit the notation and assumptions used in Sections 2 and 4. It clarifies the mathematical objects underlying recoverable observation handling and the policy objective; it does not introduce an additional optimization method.

A.1 Notation and domains

Let $\mathcal{V}$ be a finite token vocabulary, and let $\mathcal{V}^*$ denote the set of all finite token sequences over $\mathcal{V}$. A model history $h \in \mathcal{V}^*$ is the complete token sequence visible to the policy at the time an action token is generated. The policy is a family of conditional distributions $\pi_\theta(\cdot \mid h)$ over $\mathcal{V}$, parameterized by $\theta \in \Theta \subseteq \mathbb{R}^d$, where $d \geq 1$.

Table 5 summarizes the notation used throughout the paper.

Table 5: Notation used in the mathematical description. Token and character lengths are distinct quantities and are measured under fixed tokenization and character-counting conventions.

Notation	Definition
$\mathcal{V}$	Finite token vocabulary.
$\mathcal{V}^*$	Set of finite token sequences over $\mathcal{V}$.
x	Task prompt.
h	Finite history visible to the policy.
$\theta \in \Theta \subseteq \mathbb{R}^d$	Current policy parameters.
$\theta_{\text{old}} \in \Theta$	Parameters of the policy used to generate the sampled trajectory.
$\pi_\theta(v \mid h)$	Conditional probability of token $v \in \mathcal{V}$ given history h.
$\ell_{\text{tok}}(o)$	Token length of text o under the fixed tokenizer.
$\ell_{\text{char}}(o)$	Character length of text o under the fixed counting convention.
B_{obs}	Maximum amount of an individual observation retained directly in the active context.
B_{ctx}	Model context limit.
B_{ret}	Maximum amount of text exposed by one recall operation.
B_{active}	Maximum total recalled content simultaneously retained in the active context.
n	Number of sampled trajectories for one task, with $n \geq 2$.
r_i	Finite terminal reward of trajectory i.
$\bar{r}$, σ_r	Group reward mean and standard deviation defined in Equation 2.
A_i	Group-relative advantage of trajectory i.
T_i	Number of post-prompt token occurrences in trajectory i.
$\mathcal{A}_i$	Set of token positions in trajectory i generated by the policy.
y_{it}	Policy-generated token at position $t \in \mathcal{A}_i$.
h_{it}	History visible when y_{it} was generated.
$\rho_{it}(\theta)$	Likelihood ratio defined in Equation 6.
w_{it}	Nonnegative aggregation weight for a generated token.
$\mathcal{L}_{it}(\theta, A_i)$	Per-token policy-objective term determined by the underlying RL algorithm.
$J_x(\theta)$	Reward-dependent sampled objective associated with task x.

The distinction between token length and character length is important. A bound expressed in characters does not, by itself, imply a corresponding bound in tokens. Similarly, B_{ret} and B_{active} con-

strain the recall interface, whereas B_{ctx} constrains the complete model input. These quantities therefore represent different limitations and should not be used interchangeably.

A.2 Group-relative advantages

For a fixed task x, suppose that $n \geq 2$ trajectories are sampled with finite terminal rewards $r_1, \ldots, r_n \in \mathbb{R}$. We define

$$\bar{r} = \frac{1}{n}\sum_{i=1}^{n} r_i, \qquad \sigma_r = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(r_i - \bar{r})^2}, \qquad A_i = \frac{r_i - \bar{r}}{\sigma_r + \delta}, \tag{8}$$

where $\delta > 0$ is a fixed stabilizing constant.

The denominator is strictly positive, so A_i is well defined for every finite reward vector. If all trajectories receive the same reward, then $\sigma_r = 0$ and $A_i = 0$ for every i. This observation concerns the group-relative reward term only. It does not imply that the complete optimizer update vanishes, because other losses, regularizers, other task groups, or optimizer state may also affect the parameter update.

Conversely, $\sigma_r > 0$ guarantees only that the normalized advantages are not all zero. It does not guarantee that the aggregate policy gradient is nonzero, because contributions from different trajectories or token positions may cancel. We therefore use reward variation only as a property of the sampled group, not as a direct measure of gradient magnitude.

A.3 Action positions and conditioning histories

For trajectory i, let its post-prompt token occurrences be indexed by $\{1, \ldots, T_i\}$, where $T_i < \infty$. The set $\mathcal{A}_i \subseteq \{1, \ldots, T_i\}$ contains exactly those positions whose tokens were generated by the policy.

Tokens supplied by the environment are not members of $\mathcal{A}_i$. This includes ordinary tool observations, compact observation references inserted by the environment, and text returned through the recall interface. These tokens may nevertheless appear in the conditioning history of later policy-generated tokens.

For each $t \in \mathcal{A}_i$, let $y_{it} \in \mathcal{V}$ be the generated token and let $h_{it} \in \mathcal{V}^*$ be the exact history visible when that token was sampled. The history h_{it} is fixed by the rollout and is not reconstructed retrospectively during optimization.

The likelihood ratio is

$$\rho_{it}(\theta) = \frac{\pi_\theta(y_{it} \mid h_{it})}{\pi_{\theta_{\mathrm{old}}}(y_{it} \mid h_{it})}. \tag{9}$$

We assume $\pi_{\theta_{\mathrm{old}}}(y_{it} \mid h_{it}) > 0$ for every indexed generated token. This condition holds mathematically for a token sampled from the stated rollout distribution whenever its sampled probability is strictly positive, and it is required for the ratio to be well defined.

Both the numerator and denominator in Equation 9 must be evaluated on the same history h_{it}. In particular, information recalled after position t cannot be inserted retroactively into that history, and information that was visible when y_{it} was generated cannot be removed from it when recomputing likelihoods.

A.4 Action-only policy objective

The generic objective in Section 4.2 is

$$J_x(\theta) = \sum_{i=1}^{n} \sum_{t \in \mathcal{A}_i} w_{it} \mathcal{L}_{it}(\theta, A_i), \tag{10}$$

where $w_{it} \geq 0$ is finite and $\mathcal{L}_{it}$ is the per-token contribution specified by the underlying policy-optimization algorithm.

Equation 10 is intentionally algorithm-agnostic. For a likelihood-ratio-based method, $\mathcal{L}_{it}$ may depend on $\rho_{it}(\theta)$, A_i, and algorithm-specific clipping or scaling terms. The recoverable observation mechanism does not require a particular functional form for $\mathcal{L}_{it}$.

The essential requirement is instead the index set $\mathcal{A}_i$: likelihood-based policy terms are assigned only to tokens generated by the policy. Environment-generated tokens do not receive direct policy likelihood terms.

This exclusion should not be interpreted as removing observations from the computation graph of subsequent decisions. If an environment observation is part of h_{it}, then it conditions $\pi_\theta(y_{it} \mid h_{it})$ and can therefore affect the gradient associated with the later action. The distinction is between *conditioning information* and *prediction targets*: observations belong to the former but not the latter.

A.5 Regularization and auxiliary losses

The reward-dependent objective in Equation 10 need not be the complete training objective. For example, a policy-optimization method may include a regularization term that discourages deviation from a reference policy.

For probability distributions P and Q over $\mathcal{V}$, the Kullback–Leibler divergence is

$$D_{\mathrm{KL}}(P\|Q) = \sum_{v\in\mathcal{V}:\, P(v)>0} P(v)\log\frac{P(v)}{Q(v)}, \tag{11}$$

provided that $Q(v) > 0$ whenever $P(v) > 0$. If this support condition is violated, the divergence is infinite.

More generally, let $K_x : \Theta \to \mathbb{R}$ denote a differentiable auxiliary term and let $\beta \geq 0$. A complete sampled loss may have the form

$$L_x(\theta) = -J_x(\theta) + \beta K_x(\theta). \tag{12}$$

Hence, even when J_x is locally constant or has zero gradient for a particular sampled group, $\nabla_\theta L_x(\theta)$ need not vanish. The parameter update may also include contributions from other task groups, weight decay, or optimizer state. Statements about the reward-dependent objective should therefore not be interpreted as statements about the complete parameter update unless all additional terms are explicitly accounted for.

A.6 Recoverability and information access

The recall constraints in Equation 5 regulate information exposed to the model, not the size of the external archive. In particular, storing an observation in full does not imply that the entire observation can be presented to the policy in a single recall.

Let o denote an archived observation and let $R(o)$ denote the text returned by one recall operation. The recall interface requires

$$\ell_{\mathrm{char}}(R(o)) \leq B_{\mathrm{ret}}. \tag{13}$$

If multiple recalled texts $u_1, \ldots, u_q$ are simultaneously active, they additionally satisfy

$$\sum_{j=1}^{q} \ell_{\mathrm{char}}(u_j) \leq B_{\mathrm{active}}. \tag{14}$$

These bounds formalize *bounded recoverability*: information removed from the active context is not necessarily destroyed, but access to it remains constrained by the recall interface. In particular, if the return procedure exposes only selected portions of a long archived observation, preservation of the complete observation does not imply that every character is reachable through a single recall.

Identifier resolution is logically separate from these length constraints. The method assumes that a valid observation reference can be mapped to the intended archived observation. Collision handling and identifier construction are implementation properties and should be specified separately when they affect this assumption.

B Full Hyperparameter and Configuration Table

Paired training. The Agent0 and Ours runs for a given backbone are identical in every hyperparameter — model, data, batch size, learning rate, KL coefficient, rollout count, tool wiring — except the one flag. The Base and R-Zero columns are reference points, not controlled contrasts.

One deviation from Agent0 applies to *both* arms of every pair: we freeze the curriculum, building the training mixture once and holding it byte-identical across the pair, rather than letting Agent0's Curriculum/Executor co-evolution loop regenerate it each round and make the training distribution itself a function of the substrate under test. Our "Agent0" arm is therefore Agent0's executor and ADPO objective on a fixed mixture, not the full co-evolutionary system, and its absolute scores should not be read as a reproduction of Xia et al. (2026).

Three inference conditions.

Comparing checkpoints only with the mechanism switched on would conflate what *training with* the mechanism changes and what *having access to* it at inference changes independent of training; a third question hides inside the second, since the text describing the recall protocol is itself a prompt intervention. We therefore evaluate under three inference conditions — **truncate** (Agent0), **recall** (Ours) (mechanism available, protocol disclosed in the system prompt), and **recall-silent** (Ours) (mechanism available, protocol *not* disclosed) — yielding, per backbone, the grid:

Method	**truncate**	**recall (Ours)**	**recall-silent (Ours)**
Base (untrained)	A	B	C
R-Zero	R	—	—
Agent0-trained	D	E	F
Ours-trained	G	H	I

The **row effect** ($G-D$, $H-E$, $I-F$) isolates what training with the mechanism bought, holding the inference harness fixed: the paper's training-time claim. The **column effect** ($B-A$, $C-A$, etc.) isolates what the mechanism buys at inference on a fixed-training policy, and is answerable even from an untrained model. The **recall** versus **recall-silent** contrast separates instruction-following from an internalized policy for when to use the mechanism: if an untrained model matches a trained checkpoint once the protocol is disclosed, no disclosed-condition result can be attributed to training.

Coverage gaps, stated up front. The grid is complete for the Base, Agent0 and Ours rows on LongBench-v2 (40 cells including R-Zero). On the math suite, Base and R-Zero were evaluated only under **truncate** (32 cells), so the math column effect is measurable on the two agentically-trained rows but not the untrained control; R-Zero is evaluated only under **truncate** throughout, since the recall protocol has no counterpart in its non-agentic harness.

Backbones. Four models from two families, trained and evaluated independently: **Qwen3-4B** and **Qwen3-8B** (base pretrained) and **Gemma-3-4B-it** and **Gemma-2-9B-it** (instruction-tuned). The instruction-tuned pair tests whether the training-signal intervention survives on policies that already follow protocol instructions out of the box; §6.3 finds only a partial yes. Runs are never step-matched across backbones; all comparisons are within-backbone.

Training configuration. ADPO (Xia et al., 2026): $s_{\min} = 0.3$, $s_{\max} = 0.8$, $\alpha_{\min} = 0.6$, $\varepsilon_{\text{bonus}} = 0.1$; rollout group size $n = 8$, train and mini-batch sizes of 16; maximum prompt/response lengths of 1,024/4,096 tokens, at most 6 turns per rollout. The observation budget is 512 tokens, serving as both Agent0's truncation threshold and our citation trigger. KL coefficient 10^{-2}, actor learning rate 10^{-6}. Rollouts run synchronously and in-process due to an incompatibility between the framework's asynchronous path (Sheng et al., 2025) and our pinned inference library. Two tools are exposed: Agent0's Python interpreter and a memory-stress tool that serves the target on the first request and a decoy afterward, preventing trivial re-requests. Runs use 100 optimizer steps (2–3 min/step), ≈7–10 hours (30–40 GPU-hours) per matched pair on 4×A100 80GB. Recall is capped at 16,000 characters/trajectory, 8,000/recall, 5 recall actions/turn (full configuration in Appendix B).

Training uses only terminal outcome rewards, $+1/-1$ for a correct/incorrect extracted answer ($r = 2a - 1$), with no process reward or recall-action shaping; memory-stress tasks use the same rule. A

structurally unsolvable row therefore produces a zero-variance reward group — every rollout scores -1 — and no gradient.

Data. A fixed mixture, byte-identical across the two methods of each pair: real math problems scored by the self-consistency procedure Agent0's curriculum-curation pipeline uses (Wang et al., 2022), plus the synthetic memory-stress task of §5.2 at roughly 14% of rows. The math rows are identical between methods; the methods differ only in one system-prompt line describing the recall protocol, present only in our method.

Table 6: Hyperparameters and configuration.

Knob	Value used in reported runs
GPUs	$4\times$A100-80GB
Backbones	Qwen3-4B, Qwen3-8B, Gemma-3-4B-it, Gemma-2-9B-it
RL algorithm	ADPO (advantage-scaling + clip-widening GRPO variant)
$s_{\min}$, $s_{\max}$ (ADPO score band)	0.3, 0.8
$\alpha_{\min}$ (min advantage scale)	0.6
$\varepsilon_{\text{bonus}}$ (max epsilon bonus)	0.1
Rollout group size n	8
Train / PPO-mini batch size	16
Max prompt length	1024 tokens
Max response length	4096 tokens
Max turns per rollout	6
Max model length	Qwen3-4B/8B: 10,240; Gemma-4B/9B: 8,192
Observation truncation budget	512 tokens
Recall total budget per trajectory	16,000 characters (all runs)
Recall per-action cap	8,000 characters (all runs)
Max recall actions per turn	5
KL loss coefficient	1×10^{-2}
Actor learning rate	1×10^{-6}
Entropy coefficient	0
Rollout mode	synchronous, in-process
Optimizer steps per run	100
Validation cadence	every 20–25 steps
Wall-clock per step	2–3 minutes
Wall-clock per matched pair	7–10 hours on $4\times$A100 ($\approx$30–40 GPU-hours)
Evaluation: LongBench-v2	max model length=8192, max tokens=2048, $N = 311$
Evaluation: math tasks	max model length=8192, max tokens=4096, $N = 2{,}866$

C Full LongBench-v2 Results

$N = 311$, 4-way multiple choice, chance $= 25.0\%$. Accuracy counts no-answer items as incorrect; "acc | ans" divides by answered items only. z is against the 25.0% chance floor; the Bonferroni threshold over 40 cells is $p < 0.00125$ and BH-FDR at 0.05 rejects only the Qwen3-8B Base **truncate**.

Pooled over the 12 complete-grid checkpoint cells ($n = 3{,}732$ per condition): accuracy 23.93% (**truncate**), 26.31% (**recall-silent (Ours)**), 26.34% (**recall (Ours)**); accuracy conditional on answering 26.81%, 27.29%, 28.08%; abstention 10.74%, 3.59%, 6.19%. The pooled **recall-silent (Ours)** minus **truncate** accuracy gain of $+2.38$ pp decomposes into $+1.92$ pp attributable to the abstention reduction at unchanged conditional accuracy and $+0.46$ pp attributable to the 0.48-point rise in conditional accuracy: five sixths of the visible accuracy movement is a denominator effect. Pooled by training method at **truncate** (4 backbones, $n = 1{,}244$): Base 19.77% (19.53% abstention), R-Zero 26.37% (3.94%), Agent0 25.72% (7.32%), Ours 26.29% (5.39%).

The per-backbone rows in Table 7 show this pattern does not hold uniformly. The two Qwen3 backbones carry the effect: the table's highest accuracy, $98/311$ ($z = +2.65$), is reached twice, once by Qwen3-4B/Ours/**recall (Ours)** and once by Qwen3-8B/*Agent0*/**recall-silent (Ours)**. That the

Table 7: Full LongBench-v2 Implementation Results.

Model	Train-Method	Inference	Pass	Acc %	Wilson 95% CI	No-ans	Acc| ans	z vs. chance
Qwen3-4B	Base	truncate	76	24.44	[20.0, 29.5]	27	26.76	−0.23
Qwen3-4B	Base	recall-silent	88	28.30	[23.6, 33.5]	1	28.39	+1.34
Qwen3-4B	Base	recall	80	25.72	[21.2, 30.9]	12	26.76	+0.29
Qwen3-4B	R-Zero	truncate	81	26.05	[21.5, 31.2]	0	26.05	+0.43
Qwen3-4B	Agent0	truncate	82	26.37	[21.8, 31.5]	18	27.99	+0.56
Qwen3-4B	Agent0	recall-silent	95	30.55	[25.7, 35.9]	1	30.65	+2.26
Qwen3-4B	Agent0	recall	91	29.26	[24.5, 34.5]	1	29.35	+1.74
Qwen3-4B	Ours	truncate	88	28.30	[23.6, 33.5]	3	28.57	+1.34
Qwen3-4B	Ours	recall-silent	92	29.58	[24.8, 34.9]	2	29.77	+1.87
Qwen3-4B	Ours	**recall**	**98**	**31.51**	[26.6, 36.9]	8	**32.34**	+2.65
Qwen3-8B	Base	truncate	49	15.76	[12.1, 20.2]	**149**	30.25	−3.76
Qwen3-8B	Base	recall-silent	74	23.79	[19.4, 28.8]	15	25.00	−0.49
Qwen3-8B	Base	recall	73	23.47	[19.1, 28.5]	21	25.17	−0.62
Qwen3-8B	R-Zero	truncate	86	27.65	[23.0, 32.9]	0	27.65	+1.08
Qwen3-8B	Agent0	truncate	90	28.94	[24.2, 34.2]	0	28.94	+1.60
Qwen3-8B	Agent0	**recall-silent**	**98**	**31.51**	[26.6, 36.9]	4	31.92	+2.65
Qwen3-8B	Agent0	recall	95	30.55	[25.7, 35.9]	18	**32.42**	+2.26
Qwen3-8B	Ours	truncate	93	29.90	[25.1, 35.2]	1	30.00	+2.00
Qwen3-8B	Ours	recall-silent	97	31.19	[26.3, 36.5]	5	31.70	+2.52
Qwen3-8B	Ours	recall	89	28.62	[23.9, 33.9]	4	28.99	+1.47
Gemma-3-4B-it	Base	truncate	62	19.94	[15.9, 24.7]	28	21.91	−2.06
Gemma-3-4B-it	Base	recall-silent	76	24.44	[20.0, 29.5]	4	24.76	−0.23
Gemma-3-4B-it	Base	recall	74	23.79	[19.4, 28.8]	17	25.17	−0.49
Gemma-3-4B-it	R-Zero	truncate	86	27.65	[23.0, 32.9]	4	28.01	+1.08
Gemma-3-4B-it	Agent0	truncate	83	26.69	[22.1, 31.9]	11	27.67	+0.69
Gemma-3-4B-it	Agent0	recall-silent	79	25.40	[20.9, 30.5]	3	25.65	+0.16
Gemma-3-4B-it	Agent0	recall	84	27.01	[22.4, 32.2]	1	27.10	+0.82
Gemma-3-4B-it	Ours	truncate	87	27.97	[23.3, 33.2]	8	28.71	+1.21
Gemma-3-4B-it	Ours	recall-silent	82	26.37	[21.8, 31.5]	0	26.37	+0.56
Gemma-3-4B-it	Ours	**recall**	**90**	**28.94**	[24.2, 34.2]	3	**29.22**	+1.60
Gemma-2-9B-it	Base	truncate	59	18.97	[15.0, 23.7]	39	21.69	−2.46
Gemma-2-9B-it	Base	recall-silent	66	21.22	[17.0, 26.1]	17	22.45	−1.54
Gemma-2-9B-it	Base	recall	66	21.22	[17.0, 26.1]	49	25.19	−1.54
Gemma-2-9B-it	R-Zero	truncate	75	24.12	[19.7, 29.2]	45	28.20	−0.36
Gemma-2-9B-it	Agent0	truncate	65	20.90	[16.8, 25.8]	**62**	26.10	−1.67
Gemma-2-9B-it	Agent0	**recall-silent**	**75**	**24.12**	[19.7, 29.2]	43	**27.99**	−0.36
Gemma-2-9B-it	Agent0	recall	69	22.19	[17.9, 27.1]	55	26.95	−1.15
Gemma-2-9B-it	Ours	truncate	59	18.97	[15.0, 23.7]	55	23.05	−2.46
Gemma-2-9B-it	Ours	recall-silent	60	19.29	[15.3, 24.0]	39	22.06	−2.32
Gemma-2-9B-it	Ours	recall	74	23.79	[19.4, 28.8]	42	27.51	−0.49

joint-best cell is shared with an Agent0-trained checkpoint is worth stating plainly: on LongBench-v2 the inference condition matters more than which of the two methods produced the checkpoint. Gemma-2-9B-it is a null backbone throughout: all nine of its checkpoint × inference-condition cells have negative z vs. chance, and Ours training does not close the gap ($z = -2.46$ under **truncate** for both Base and Ours). Qwen3-8B/Base/**truncate** is the one cell that is significantly *below* chance, driven by a 48% abstention rate ($149/311$ no-answer) rather than wrong answers among those it did answer (Acc|ans 30.25%, above chance).

Table 8 gives the full per-cell breakdown underlying §6.2's condensed Table 2, with *z*/*p* comparing **recall-silent** against **truncate** abstention for each of the 12 complete-grid checkpoint cells.

D Full Math Evaluation Per-Benchmark Results

Per-benchmark accuracy in percent. N: GSM8K 1,319; MATH-500 500; AIME24 30; AIME25 30; AMC-23 40; Minerva 272; OlympiadBench 675; total 2,866. "micro" is the problem-count-weighted pooled average, "macro" the unweighted benchmark average, "non-GSM8K" the pooled average over the remaining 1,547 problems.

Pooled across the four models ($n = 11{,}464$ per condition), Ours-trained minus Agent0-trained: **truncate** 60.81 vs. 60.01 (+0.79 pp, $p = 0.22$); **recall-silent (Ours)** 61.39 vs. 59.22 (+2.17 pp, $p = 7.8 \times 10^{-4}$); **recall (Ours)** 61.21 vs. 59.70 (+1.50 pp, $p = 0.020$). Restricting to the non-GSM8K pool ($n = 6{,}188$): +0.28 ($p = 0.75$), +2.19 ($p = 0.013$), +1.36 ($p = 0.12$).

Table 8: LongBench-v2 abstention, z and p compare **recall-silent** against **truncate**.

Model	Method	truncate	recall-silent (Ours)	recall (Ours)	Δ	z	p
Qwen3-4B	Base	8.7% (27)	0.3% (1)	3.9% (12)	-8.4 pp	-5.13	2.9×10^{-7}
Qwen3-4B	Agent0	5.8% (18)	0.3% (1)	0.3% (1)	-5.5 pp	-4.01	6.0×10^{-5}
Qwen3-4B	Ours	1.0% (3)	0.6% (2)	2.6% (8)	-0.3 pp	-0.45	0.65
Qwen3-8B	Base	**47.9% (149)**	**4.8% (15)**	6.8% (21)	-43.1 pp	-13.98	$< 10^{-40}$
Qwen3-8B	Agent0	0.0% (0)	1.3% (4)	5.8% (18)	$+1.3$ pp	$+2.01$	0.044
Qwen3-8B	Ours	0.3% (1)	1.6% (5)	1.3% (4)	$+1.3$ pp	$+1.64$	0.10
Gemma-3-4B-it	Base	9.0% (28)	1.3% (4)	5.5% (17)	-7.7 pp	-4.42	9.7×10^{-6}
Gemma-3-4B-it	Agent0	3.5% (11)	1.0% (3)	0.3% (1)	-2.6 pp	-2.17	0.030
Gemma-3-4B-it	Ours	2.6% (8)	0.0% (0)	1.0% (3)	-2.6 pp	-2.87	0.004
Gemma-2-9B-it	Base	12.5% (39)	5.5% (17)	15.8% (49)	-7.1 pp	-3.11	0.0019
Gemma-2-9B-it	Agent0	19.9% (62)	13.8% (43)	17.7% (55)	-6.1 pp	-2.04	0.041
Gemma-2-9B-it	Ours	17.7% (55)	12.5% (39)	13.5% (42)	-5.1 pp	-1.80	0.072
Pooled ($n = 3{,}732$)		**10.7% (401)**	**3.6% (134)**	6.2% (231)	-7.2 pp	-12.10	$< 10^{-30}$

Table 9: Full Math Bechmark Implementation Results.

Model	Train	Inference	GSM8K	MATH-500	AIME24	AIME25	AMC-23	Minerva	Olympiad	ACC	macro	non-GSM8K
Qwen3-4B	Base	truncate	79.2	57.0	23.3	10.0	45.0	24.3	30.8	56.93	38.51	37.94
Qwen3-4B	R-Zero	truncate	78.8	55.0	23.3	10.0	42.5	26.5	30.7	56.55	38.11	37.58
Qwen3-4B	Agent0	truncate	81.7	58.6	**30.0**	10.0	52.5	26.1	32.7	59.15	41.66	39.93
Qwen3-4B	Agent0	re-silent	85.2	61.0	20.0	13.3	37.5	**29.0**	33.9	61.46	39.99	41.22
Qwen3-4B	Agent0	recall	80.2	57.2	20.0	**20.0**	47.5	27.9	29.2	57.50	40.29	38.14
Qwen3-4B	Ours	truncate	83.4	62.6	23.3	13.3	**57.5**	25.7	33.5	60.82	42.76	41.56
Qwen3-4B	**Ours**	**re-silent**	**85.7**	**65.8**	23.3	**20.0**	**57.5**	**29.0**	**37.2**	**63.69**	**45.50**	**44.92**
Qwen3-4B	Ours	recall	81.9	61.2	10.0	13.3	42.5	28.3	32.0	59.43	38.46	40.27
Qwen3-8B	Base	truncate	80.1	59.0	**23.3**	6.7	50.0	27.2	29.0	57.58	39.33	38.38
Qwen3-8B	R-Zero	truncate	**90.4**	61.4	13.3	3.3	30.0	24.6	24.4	60.99	35.34	35.91
Qwen3-8B	Agent0	truncate	82.0	60.2	**23.3**	6.7	50.0	**30.9**	30.8	59.44	40.56	40.20
Qwen3-8B	Agent0	re-silent	73.0	58.0	10.0	10.0	35.0	29.0	29.6	54.14	34.94	38.05
Qwen3-8B	Agent0	recall	84.8	63.4	10.0	**13.3**	50.0	29.4	32.9	61.57	40.54	41.76
Qwen3-8B	Ours	truncate	84.8	62.2	16.7	**13.3**	**52.5**	29.4	31.0	61.02	41.41	40.74
Qwen3-8B	Ours	re-silent	81.3	61.8	13.3	10.0	35.0	29.0	32.4	59.31	37.54	40.57
Qwen3-8B	**Ours**	**recall**	87.5	**65.4**	16.7	6.7	50.0	30.5	**34.4**	**63.62**	**41.60**	**43.26**
Gem-3-4B-it	Base	truncate	89.1	70.6	6.7	13.3	50.0	19.5	35.6	64.47	40.69	43.46
Gem-3-4B-it	R-Zero	truncate	88.7	70.2	6.7	13.3	40.0	18.8	35.3	63.93	39.00	42.82
Gem-3-4B-it	Agent0	truncate	90.0	72.2	6.7	10.0	45.0	20.2	**35.7**	65.14	39.97	43.95
Gem-3-4B-it	Agent0	re-silent	89.8	**73.6**	6.7	6.7	45.0	18.4	34.7	64.86	39.27	43.59
Gem-3-4B-it	Agent0	recall	89.1	70.0	10.0	10.0	40.0	16.9	34.5	63.72	38.64	42.07
Gem-3-4B-it	Ours	truncate	**90.1**	73.6	3.3	**10.0**	45.0	19.1	34.8	65.08	39.41	43.75
Gem-3-4B-it	**Ours**	**re-silent**	**90.1**	73.4	**10.0**	16.7	**55.0**	**22.8**	34.8	**65.68**	43.26	**44.86**
Gem-3-4B-it	Ours	recall	89.8	72.8	6.7	**20.0**	42.5	15.8	35.4	64.74	40.43	43.37
Gem-2-9B-it	Base	truncate	69.1	40.0	3.3	**6.7**	17.5	19.5	17.2	45.03	24.76	24.51
Gem-2-9B-it	R-Zero	truncate	87.7	49.0	3.3	0.0	7.5	22.4	15.4	54.80	26.47	26.75
Gem-2-9B-it	Agent0	truncate	87.7	**51.8**	3.3	0.0	15.0	26.1	17.8	56.31	28.81	29.55
Gem-2-9B-it	Agent0	re-silent	88.7	50.0	0.0	0.0	17.5	26.1	17.6	56.41	28.56	28.88
Gem-2-9B-it	Agent0	recall	87.1	50.6	3.3	**6.7**	**27.5**	24.6	18.2	56.02	**31.14**	29.53
Gem-2-9B-it	Ours	truncate	88.7	49.6	0.0	0.0	20.0	26.1	17.3	56.31	28.81	28.69
Gem-2-9B-it	Ours	re-silent	88.2	50.0	**6.7**	0.0	22.5	27.6	**19.3**	56.86	30.61	**30.15**
Gem-2-9B-it	**Ours**	**recall**	88.7	51.0	3.3	3.3	17.5	**28.7**	18.2	**57.04**	30.10	30.05

The row-wise bold cells track the pooled ranking above: **recall-silent (Ours)** is the best condition for the Ours-trained checkpoint on micro-average in two of four models (Qwen3-4B 63.69, Gemma-3-4B-it 65.68), with **recall (Ours)** best for Qwen3-8B (63.62) and Gemma-2-9B-it (57.04). On the non-GSM8K pool the split is the same two-two, with **recall-silent** leading on Qwen3-4B (44.92) and Gemma-2-9B-it (30.15) and **recall** leading on Qwen3-8B (43.26) and Gemma-3-4B-it (43.37); note in particular that on Qwen3-8B **recall-silent** is the *worst* of the three non-GSM8K cells (40.57 against 40.74 and 43.26). The uniform **recall-silent** advantage reported in §7.2 therefore holds within the Qwen3-4B length sweep but does not generalize across backbones, and we do not claim it does. Qwen3-8B/R-Zero/**truncate** is the table's sharpest internal split: its GSM8K score (90.4) is the highest for that model, but AIME24 (13.3) and AIME25 (3.3) are its lowest, pulling non-GSM8K down to 35.91 — the same benchmark, AIME24, that Appendix E flags as the least stable under the Ours-vs-Agent0 contrast. Gemma-2-9B-it's accuracy trails the other three backbones by 10–20 points in every checkpoint $\times$ condition cell (45–57% vs. 55–66% elsewhere), the same backbone that showed no benefit from our training on LongBench-v2 in Appendix C — on both tasks, training-time intervention has little purchase on this backbone.

Comparison against R-Zero (Huang et al., 2026). Table 10 compares each backbone's Ours-trained checkpoint against its R-Zero checkpoint under two readings: matched at **truncate** (the only condition R-Zero can be run in) and at our model's best of three inference conditions.

Table 10: Comparing each backbone's Ours-trained model to its R-Zero checkpoint, under two readings: matched at truncate (the only condition R-Zero can be run in) and at our model's best of three conditions

Model	R-Zero	Ours	Δ	Ours (best condition)	Δ
Qwen3-4B	56.55	**60.82**	$+4.27$ ($p = 0.001$)	63.69 (recall-silent)	$+7.14$ ($p < 10^{-4}$)
Qwen3-8B	60.99	**61.02**	$+0.03$ ($p = 0.98$)	63.62 (recall)	$+2.63$ ($p = 0.040$)
Gemma-3-4B-it	63.93	**65.08**	$+1.15$ ($p = 0.36$)	65.68 (recall-silent)	$+1.74$ ($p = 0.17$)
Gemma-2-9B-it	54.80	**56.31**	$+1.50$ ($p = 0.25$)	57.04 (recall)	$+2.24$ ($p = 0.088$)
Pooled accuracy ($n = 11{,}464$)	59.07	**60.81**	$+1.74$ ($p = 0.007$)	62.51	$+3.44$ ($p < 10^{-7}$)
Pooled non-GSM8K ($n = 6{,}188$)	35.77	**38.69**	$+2.92$ ($p = 0.0008$)	40.77	$+5.01$ ($p < 10^{-8}$)

The matched-condition column shows Ours ahead on all four backbones, decisively on Qwen3-4B and by a hair on Qwen3-8B. The non-GSM8K pooled gap is larger than the micro gap in every case, for the reason given in §6.3. The best-condition column is an oracle over the three inference conditions.

E Per-Benchmark Direction of the Training-Time Effect

Direction of Ours-minus-Agent0 contrast across the 12 matched comparisons (4 backbones $\times$ 3 inference conditions), broken out by benchmark. Ties are exact equalities in the logged accuracy.

Table 11: Direction of the Ours-minus-Agent0 contrast, by benchmark, over the 12 matched comparisons.

Benchmark	N	improves	regresses	ties
GSM8K	1,319	11	1	0
MATH-500	500	9	2	1
OlympiadBench	675	9	2	1
AMC-23	40	7	2	3
AIME25	30	5	3	4
AIME24	30	5	6	1
Minerva	272	5	4	3
Total		**51**	**20**	**13**

The effect is carried by the three largest benchmarks and is absent or reversed on Minerva and AIME24. The AIME sets have 30 problems each, so one item is 3.3 percentage points and their per-cell directions carry almost no information; Minerva's near-null is a genuine null on 272 problems and we have no account of it.

Breaking the AIME24 row out by inference condition (Appendix D) shows the split is not random with respect to condition: **recall-silent (Ours)** is the only one of the three where Ours improves over Agent0 in all four backbones, while the six regressions fall entirely under **truncate** (4 of 4 backbones) and **recall (Ours)** (2 of 4). This within-benchmark pattern runs in the same direction as the Qwen3-4B length sweep of §7.2, in which **recall-silent (Ours)** is the best inference condition in all 27 cells. We note, however, that the **recall-silent** advantage is uniform only *within* that single-backbone sweep: across the four backbones of the main grid it holds in two of four (Appendix D).

F Full Seed-Robustness Results

Table 12 gives the full per-benchmark math-suite results underlying §7.1's seed-robustness check, mean over three training seeds (42, 82, 122).

Table 12: Math-task per-benchmark accuracy on Qwen3 models, mean over three training seeds (42, 82, 122); ±SD is given on the pooled column. AVG_ACC is the seven-benchmark pooled accuracy.

Model	Method	Inference	GSM8K	MATH-500	AIME24	AIME25	AMC-23	Minerva	Olympiad	AVG_ACC
Q3-4B	Agent0	truncate	79.67	57.40	20.00	12.20	43.33	25.50	31.20	57.40 ± 0.50
Q3-4B	Agent0	re-silent	82.67	59.13	16.67	10.00	40.00	27.60	34.83	60.03 ± 0.15
Q3-4B	Agent0	recall	79.10	57.33	12.20	11.10	40.00	26.70	28.50	56.43 ± 0.35
Q3-4B	Ours	truncate	81.90	60.13	**22.20**	**13.33**	**53.33**	25.10	32.67	59.37 ± 0.12
Q3-4B	**Ours**	**re-silent**	**85.17**	**63.67**	21.13	11.10	51.67	**29.00**	**35.40**	$\mathbf{62.47} \pm 0.40$
Q3-4B	Ours	recall	80.90	59.07	12.23	8.90	45.83	26.60	31.23	58.27 ± 0.25
Q3-8B	Agent0	truncate	78.47	58.73	16.70	6.70	43.33	28.17	29.47	56.80 ± 1.39
Q3-8B	Agent0	re-silent	67.03	58.87	12.20	7.77	**44.17**	27.80	30.80	51.83 ± 1.95
Q3-8B	Agent0	recall	82.33	62.33	6.67	6.67	40.00	28.30	30.50	59.33 ± 0.83
Q3-8B	Ours	truncate	82.30	59.33	18.90	7.77	47.50	29.03	**32.40**	59.57 ± 0.35
Q3-8B	Ours	re-silent	80.77	60.40	**20.00**	**11.13**	43.33	**30.00**	32.30	59.10 ± 2.25
Q3-8B	**Ours**	**recall**	**84.10**	**64.67**	8.90	11.10	42.50	27.93	32.00	$\mathbf{61.00} \pm 0.26$

G Full Length-Budget Ablation Results (Qwen3-4B)

Per-benchmark accuracy in percent, for every (training-time context ceiling × training-time max-response length × eval-time max-response length × inference condition) combination that was run for the Qwen3-4B Ours-trained checkpoints of §7.2. Benchmark N as in Appendix D; "Accuracy" is the problem-count-weighted pooled average over all seven benchmarks. All three inference conditions in this table are the same Ours-trained checkpoint; no Agent0-trained checkpoint appears. The trajectory-level recall budget (16,000 characters) and per-recall cap (8,000 characters) are held at the Appendix B values throughout.

Table 13: Full Length-Budget Ablation Results

Train model-len.	Train max-resp.	Eval max-resp.	Inference	GSM8K	MATH 500	AIME24	AIME25	AMC-23	Minerva	Olympiad	ACC
8,192	2,048	2,048	truncate	81.7	57.2	**23.3**	**16.7**	45.0	26.1	30.2	58.2
8,192	2,048	2,048	**recall-silent**	**84.8**	**64.2**	13.3	6.7	**47.5**	26.5	**33.6**	**61.5**
8,192	2,048	2,048	recall	80.1	58.8	16.7	10.0	35.0	**28.3**	30.4	57.7
8,192	3,072	3,072	truncate	81.3	59.0	**20.0**	**16.7**	45.0	23.9	30.1	58.1
8,192	3,072	3,072	**recall-silent**	**84.0**	**59.4**	10.0	10.0	**47.5**	**27.2**	**33.6**	**60.4**
8,192	3,072	3,072	recall	78.8	58.8	16.7	10.0	45.0	26.8	27.3	56.4
8,192	4,096	2,048	truncate	82.5	63.0	**30.0**	10.0	47.5	25.4	32.6	60.1
8,192	4,096	2,048	**recall-silent**	**85.6**	**65.0**	26.7	13.3	**52.5**	**30.5**	**37.0**	**63.5**
8,192	4,096	2,048	recall	82.0	60.8	13.3	**20.0**	**52.5**	27.6	31.6	59.5
8,192	4,096	4,096	truncate	83.4	62.6	**23.3**	13.3	**57.5**	25.7	33.5	60.8
8,192	4,096	4,096	**recall-silent**	**85.7**	**65.8**	**23.3**	**20.0**	**57.5**	**29.0**	**37.2**	**63.7**
8,192	4,096	4,096	recall	81.9	61.2	10.0	13.3	42.5	28.3	32.0	59.4
10,240	2,048	1,024	truncate	79.4	56.4	**23.3**	**13.3**	**52.5**	25.0	31.1	57.2
10,240	2,048	1,024	**recall-silent**	**82.9**	**58.6**	16.7	10.0	47.5	**27.6**	**32.4**	**59.6**
10,240	2,048	1,024	recall	78.4	57.0	13.3	10.0	42.5	25.4	27.1	55.7
10,240	2,048	2,048	truncate	79.8	57.6	**23.3**	**13.3**	**50.0**	24.3	30.8	57.4
10,240	2,048	2,048	**recall-silent**	**82.7**	**61.0**	16.7	6.7	42.5	**28.3**	**32.4**	**59.9**
10,240	2,048	2,048	recall	77.3	57.6	16.7	10.0	42.5	26.8	27.9	55.6
10,240	4,096	1,024	truncate	82.6	59.4	**33.3**	**13.3**	**52.5**	25.4	31.7	59.5
10,240	4,096	1,024	**recall-silent**	**85.7**	**63.2**	23.3	10.0	42.5	**29.0**	**35.9**	**62.6**
10,240	4,096	1,024	recall	81.5	59.6	10.0	6.7	50.0	26.5	28.9	58.1
10,240	4,096	2,048	truncate	82.3	58.8	23.3	**10.0**	47.5	26.1	31.7	59.1
10,240	4,096	2,048	**recall-silent**	**85.1**	**62.8**	**26.7**	6.7	47.5	**29.0**	**35.6**	**62.3**
10,240	4,096	2,048	recall	80.9	59.2	10.0	**10.0**	**50.0**	24.6	28.9	57.6
10,240	4,096	4,096	truncate	83.4	59.0	**23.3**	**10.0**	**55.0**	25.7	32.3	59.8
10,240	4,096	4,096	**recall-silent**	**85.1**	**62.6**	**23.3**	**10.0**	45.0	**28.3**	**35.4**	**62.1**
10,240	4,096	4,096	recall	81.0	58.6	10.0	6.7	37.5	26.5	29.3	57.6

Bold marks the best inference condition's micro-average within each of the 9 training-configuration groups; **recall-silent (Ours)** is bold in all 9, consistent with the ordering **recall-silent (Ours)** > **truncate** > **recall (Ours)** holding in every one of the 27 rows without exception (see §7.2).

The training-time max-response length effect is not monotonic at every context ceiling: at max model length = 8,192, the middle setting (3,072 tokens) underperforms both its neighbors on **recall (Ours)** (56.4, vs. 57.7 at 2,048 and 59.4–59.5 at 4,096) and on **truncate** (58.1, vs. 58.2 and 60.1–60.8), so the +2-point gain from raising training-time max-response quoted in §7.2 is measured

between the 2,048 and 4,096 endpoints and does not describe a smooth trend through 3,072. No 3,072-token run exists at max model length = 10,240, so it is not possible to tell from this grid whether the dip is specific to the smaller context ceiling.

H Full Length-Budget Ablation Results, LongBench-v2 (Qwen3-4B)

LongBench-v2 results ($N = 311$; 4-way multiple choice, chance = 25.0%) for every (training-time context ceiling × training-time max-response length × eval-time max-response length × inference condition) combination that was run for the Qwen3-4B Ours-trained checkpoints of §7.2. "Pass" is the raw count of correctly-answered items; "No-ans" is the count yielding no extracted answer (abstention), also scored as incorrect in Accuracy, matching the convention in Appendix C. The trajectory-level recall budget (16,000 characters) and per-recall cap (8,000 characters) are held at the Appendix B values throughout.

Bold marks the best inference condition's accuracy within each of the 12 training-configuration × eval-budget groups (ties bolded on both sides). **recall (Ours)** is bold or tied-bold in 8 of 12 groups, **recall-silent (Ours)** in 4, and **truncate** in 2; see Table 8 for the abstention-based account of where the ordering breaks down.

The abstention spike described qualitatively in §6.2 is visible in the No-ans column as three isolated outliers: the **recall (Ours)** rows at eval-time max-response= $4{,}096$ (43, 43, and 44 no-answer items, for the (8,192; 4,096), (10,240; 4,096), and (10,240; 2,048) training configurations respectively) are the only **recall (Ours)** cells in the table above 31; every other **recall (Ours)** row has 13 or fewer. In the two groups where **truncate** wins outright — (10,240; 4,096, eval= $2{,}048$) and (10,240; 4,096, eval= $4{,}096$) — **recall (Ours)** abstains on 7 and 43 items against **truncate**'s 21 and 26. Only the second of these is an abstention story: at eval= $4{,}096$ disclosed recall abstains $1.65\times$ as often as **truncate**, the highest such ratio in the table. At eval= $2{,}048$ disclosed recall actually abstains *less* than **truncate** and still loses, so **truncate**'s win there is not explained by the abstention spike and we do not have an account of it.

I RL Training comparison between Ours and Agent0

Table 15 gives the full set of training-dynamics metrics underlying §6.1's condensed Table 1, and Figures 2–5 give the corresponding per-backbone training curves.

In Figure 2: Nine training-time diagnostics for our method versus Agent0 over the matched 100-step run under identical budgets 16,000 chars, max recall 8,000 chars, model length 10,240; max response 4,096; dense curves are EMA-smoothed ($\alpha = 0.3$), raw per-step values shown faint. Panels, left to right by row: training reward, training accuracy , held-out validation math reward at six evaluation points; tool actions per trajectory , policy-gradient loss, KL to the reference policy; policy entropy, gradient norm, and mean response length. On the two quantities computed identically for both arms — training reward and accuracy — the curves are nearly indistinguishable (first→last: $0.469 \to 0.422$ Ours vs. $0.438 \to 0.391$ Agent0; accuracy: $0.734 \to 0.711$ vs. $0.719 \to 0.695$), confirming the two runs are matched on everything the substrate does not itself change. They separate sharply on quantities the recall channel touches directly. Tool actions per trajectory grow for Ours from parity with Agent0 to roughly $4\times$ its plateau by step 80, tracking the emergence of recall calls (2,255/15,604 rollout actions were recalls over the full run, 14.45%; recall rate climbs from 0 to a 0.33–0.46 range once the action becomes useful, per the per-step log). KL to the reference policy grows an order of magnitude further for Ours than for Agent0 ($0.0007 \to 0.031$, peak 0.066, vs. $0.0008 \to 0.005$, peak 0.007), and mean response length grows more ($1{,}059 \to 1{,}341$ tokens vs. $1{,}135 \to 1{,}242$), both consistent with the policy spending additional tokens and additional policy movement on learning when and how to invoke recall rather than answer directly. Policy-gradient loss shrinks in magnitude for both arms as training proceeds ($-0.119 \to -0.021$ Ours, $-0.132 \to -0.067$ Agent0), and policy entropy is consistently lower for Ours than Agent0 after step 40, consistent with recall giving the policy a concrete alternative to guessing under truncation rather than leaving it to spread probability mass over an unresolvable observation. The gap that matters is the top-right panel: held-out math validation reward rises for both arms through step 60 ($0.574 \to 0.644$ Ours, $0.562 \to 0.634$ Agent0) but only Ours continues to improve afterward, while

Table 14: Full Length-Budget Ablation Results on Longbench-v2

Train model-len.	Train max-resp.	Eval max-resp.	Inference	Pass	No-ans	Accuracy %
8,192	2,048	1,024	truncate	81	16	26.0
8,192	2,048	1,024	recall-silent	77	0	24.8
8,192	2,048	1,024	**recall**	**91**	3	**29.3**
8,192	2,048	2,048	truncate	75	19	24.1
8,192	2,048	2,048	recall-silent	80	2	25.7
8,192	2,048	2,048	**recall**	**87**	5	**28.0**
8,192	3,072	3,072	truncate	77	17	24.8
8,192	3,072	3,072	**recall-silent**	**86**	1	**27.7**
8,192	3,072	3,072	recall	77	13	24.8
8,192	4,096	1,024	truncate	73	20	23.5
8,192	4,096	1,024	recall-silent	78	0	25.1
8,192	4,096	1,024	**recall**	**89**	0	**28.6**
8,192	4,096	2,048	truncate	78	18	25.1
8,192	4,096	2,048	recall-silent	82	2	26.4
8,192	4,096	2,048	**recall**	**86**	6	**27.7**
8,192	4,096	4,096	truncate	65	28	20.9
8,192	4,096	4,096	recall-silent	80	10	25.7
8,192	4,096	4,096	**recall**	**81**	43	**26.0**
10,240	2,048	1,024	truncate	75	18	24.1
10,240	2,048	1,024	**recall-silent**	**86**	1	**27.7**
10,240	2,048	1,024	**recall**	**86**	3	**27.7**
10,240	2,048	2,048	truncate	77	22	24.8
10,240	2,048	2,048	recall-silent	**85**	2	**27.3**
10,240	2,048	2,048	**recall**	**85**	3	**27.3**
10,240	2,048	4,096	truncate	73	31	23.5
10,240	2,048	4,096	**recall-silent**	**79**	7	**25.4**
10,240	2,048	4,096	recall	73	44	23.5
10,240	4,096	1,024	truncate	82	19	26.4
10,240	4,096	1,024	recall-silent	81	1	26.0
10,240	4,096	1,024	**recall**	**91**	1	**29.3**
10,240	4,096	2,048	**truncate**	**86**	21	**27.7**
10,240	4,096	2,048	recall-silent	80	2	25.7
10,240	4,096	2,048	recall	84	7	27.0
10,240	4,096	4,096	**truncate**	**85**	26	**27.3**
10,240	4,096	4,096	recall-silent	77	7	24.8
10,240	4,096	4,096	recall	81	43	26.0

Agent0 peaks at step 60 (0.634) and recedes to end at 0.620; Ours ends at 0.644 against Agent0's 0.620, the same +2–3 pp order of gain reported for this backbone in Table 3.

Figure 3 Same nine panels as Figure 2, same matched budgets, for the larger backbone. Training reward and accuracy again track each other closely between arms (first→last: $0 \rightarrow 0.297$ Ours vs. $0.281 \rightarrow 0.375$ Agent0; accuracy: $0.5 \rightarrow 0.648$ vs. $0.641 \rightarrow 0.688$), so the two curricula remain

Table 15: Full training dynamics, four backbones × two methods. A0 = Agent0 method; Ours = recall method. Rows marked Δ are last minus first; mem-stress reward, math val reward, actor pg-loss, recall-action rate report as "first|last" give both endpoints; "peak" and "max" are taken over all logged steps. Bold marks the method favoured within each pair on the three quantities that separate the methods in all four backbones — final math validation reward, maximum KL, and response-length growth. Memory-stress reward is bolded where the methods separate, which is three of the four backbones; on Gemma-2-9B-it neither method ever scores above zero. Em-dashes mark quantities that do not exist in the A0 method.

Metric	Q3-4B A0	Q3-4B Ours	Q3-8B A0	Q3-8B Ours	G3-4B-it A0	G3-4B-it Ours	G2-9B-it A0	G2-9B-it Ours
optimizer steps	100	100	97	100	100	100	100	100
mem-stress reward	0.00|0.00	0.02|**0.45**	0.00|0.00	0.00|**0.72**	0.00|0.00	0.52|0.02	0.00|0.00	0.00|0.00
mem-stress reward, peak	0.00	0.86	0.00	0.72	0.00	0.97	0.00	0.00
math val reward	0.56|0.62	0.57|**0.64**	0.54|0.62	0.47|**0.63**	0.66|0.69	0.67|**0.70**	0.25|0.48	0.14|**0.54**
math val reward, peak	0.63	0.64	0.62	0.63	0.72	0.71	0.51	0.54
Δ training reward	−0.05	−0.05	+0.09	+0.29	+0.08	−0.06	+0.55	+0.89
Δ training accuracy	−0.02	−0.02	+0.05	+0.15	+0.04	−0.03	+0.27	+0.45
actor pg-loss	−0.13| − 0.07	−0.12| − 0.02	−0.09| + 0.03	−0.30| + 0.03	−0.12| + 0.02	−0.04| − 0.09	−0.18| − 0.07	−0.12| + 0.06
max actor KL loss	0.007	**0.066**	0.008	**0.014**	0.084	**0.106**	0.139	**0.339**
KL loss, Ours / A0	—	9.5×	—	1.7×	—	1.3×	—	2.4×
Δ actor entropy	+0.041	−0.006	−0.002	+0.002	−0.029	+0.020	+0.058	+0.198
Δ actor grad-norm	−0.139	+0.083	−0.045	−0.274	−1.995	−1.398	−2.354	−2.810
Δ mean response tokens	+107	+**282**	+59	+**285**	−170	+**404**	+43	+**217**
recall-action rate	—	0.00|0.33	—	0.12|0.02	—	0.01|0.47	—	0.01|0.08
recall-action rate, peak	—	0.459	—	0.379	—	0.667	—	0.440
recall actions, rollouts	—	14.45%	—	11.40%	—	13.82%	—	13.16%

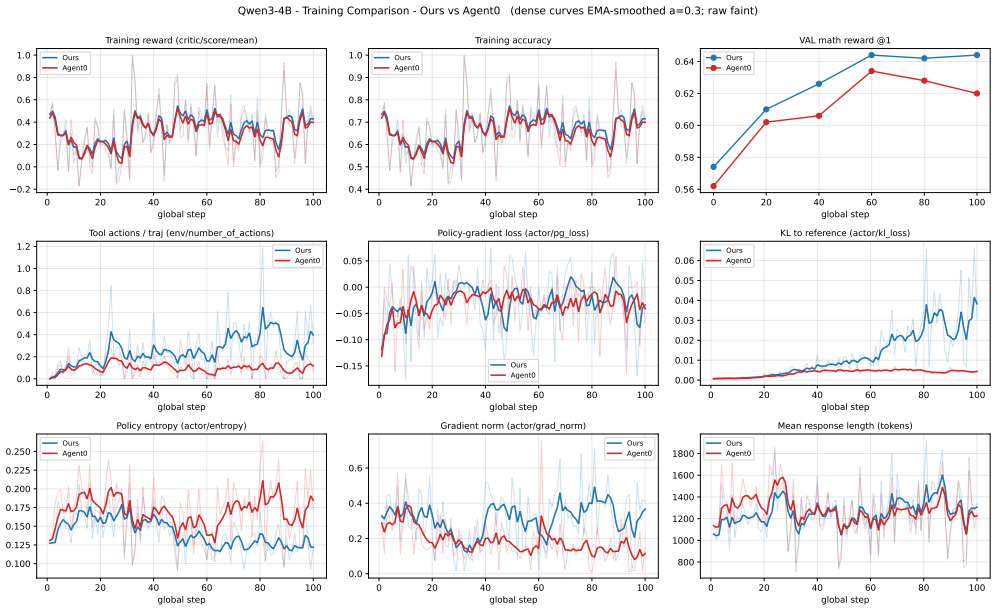

Figure 2: Qwen3-4B based-model

matched on the quantities the substrate does not itself change. Two patterns differ from the Qwen3-4B pair. First, the gap between arms on quantities the recall channel touches is present but narrower here: KL to the reference peaks at 0.0139 for Ours versus 0.0083 for Agent0 (under 2×, versus the roughly 10× separation at 4B), and mean response length shows a reversal rather than a widening gap — Agent0 starts longer (1,210 vs. 1,063 tokens) but Ours ends longer (1,348 vs. 1,269). Tool actions per trajectory are still visibly higher for Ours throughout (roughly 1.5–2× Agent0's plateau by step 80), and policy entropy is lower for Ours than Agent0 for most of the run, as at 4B. Second, recall usage is non-monotonic at this scale: recall rate opens at 0.123, peaks at 0.379 around mid-run, and closes at 0.023 (1,994/17,488 rollout actions were recalls over the full run, 11.40%) — unlike the 4B run's rate, which climbed roughly monotonically to its endpoint. Policy-gradient loss opens far more negative for Ours than Agent0 (−0.304 vs. −0.093 at step 1, reflecting a larger initial corrective update on a policy that starts further from a useful recall policy) and gradient norm mirrors

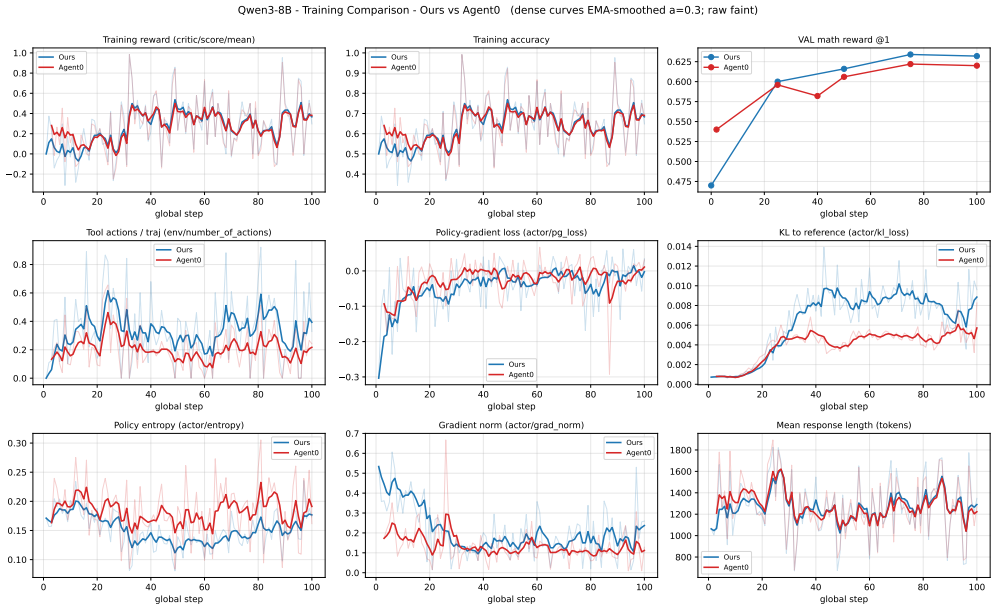

Figure 3: Qwen3-8B based-model

this (Ours opens at 0.534 against Agent0's 0.173), with both quantities converging to comparable small values by the final step (-0.021 pg-loss region and grad-norm $\approx$ 0.13–0.26 for both arms). The validation-reward panel shows the clearest backbone-specific effect: Ours *opens below* Agent0 (0.470 vs. 0.540 at step 0) but overtakes it by the second checkpoint and finishes higher (0.632 vs. 0.620), and on the memory-stress benchmark not shown in this figure, Ours reaches 0.719 by the end of training against Agent0's flat 0.

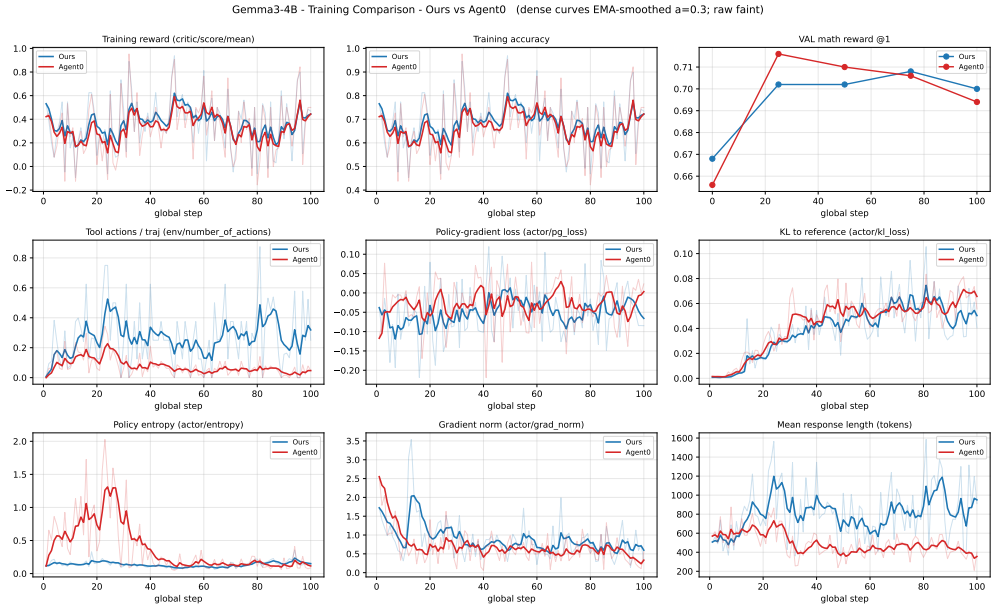

Figure 4: Gemma3-4B based-model

Figure 4 matched budgets except model length 8,192 (our chosen ceiling for the Gemma pair, versus 10,240 for the Qwen3 pair; Gemma-3 supports a far longer context natively, so this is a configuration choice and not a model limit). Training reward and accuracy track closely between arms through most of the run but diverge in the last quartile: Agent0 continues to rise slightly (mean $0.422 \rightarrow 0.500$, accuracy $0.711 \rightarrow 0.750$) while Ours declines from its own peak ($0.531 \rightarrow 0.469$, accuracy $0.766 \rightarrow 0.734$) — a -0.062 drop in training reward, and the only backbone in the study where the Ours arm declines while its paired Agent0 arm rises (§6.1; the Qwen3-4B pair also declines, but both arms decline together by an identical -0.047), visible as Ours's blue curve sagging below Agent0's red curve after step $\sim$85. Response length also moves in opposite directions for the two arms, unlike either Qwen backbone: Agent0 shrinks ($569.5 \rightarrow 399.6$ tokens) while Ours grows ($507.2 \rightarrow 911.3$ tokens), and tool actions per trajectory stay markedly higher for Ours throughout (plateauing near 0.3 versus Agent0's 0.05–0.1). The policy-entropy panel shows the sharpest cross-backbone difference in the study: Agent0's entropy spikes to a broad peak near 1.3 between steps 15–30, with no counterpart in either Qwen run, while Ours stays flat and low (≤ 0.2) throughout — the same directional gap (Ours lower) seen at both Qwen scales, but far larger and driven by a transient in Agent0 rather than a gradual separation. Gradient norm and policy-gradient loss both invert relative to the Qwen pair: Agent0's grad-norm is highest at step 1 (2.55) and falls monotonically while its pg-loss rises from -0.117 toward positive territory (0.023), whereas Ours spikes mid-run (grad-norm peak 3.54 near step 12) and its pg-loss drifts slightly more negative by the end ($-0.038 \rightarrow -0.085$) rather than toward zero. KL to the reference is the one panel where Agent0 finishes *higher* than Ours (0.056 vs. 0.041) despite Ours reaching the larger instantaneous peak (0.106 vs. 0.083) — the only backbone in the study where Agent0's final KL exceeds Ours's. Recall usage rises the most sharply of any run ($0.006 \rightarrow$ peak $0.667 \rightarrow$ final-step 0.465; 2,494/18,051 rollout actions were recalls overall, 13.82%), and memory-stress reward (not shown in this figure) tracks it almost exactly: Ours reaches 0.969 by mid-training — the highest of any backbone — then collapses to 0.016 by the final checkpoint even as recall usage keeps climbing, the "solved the task and then stopped" pattern.

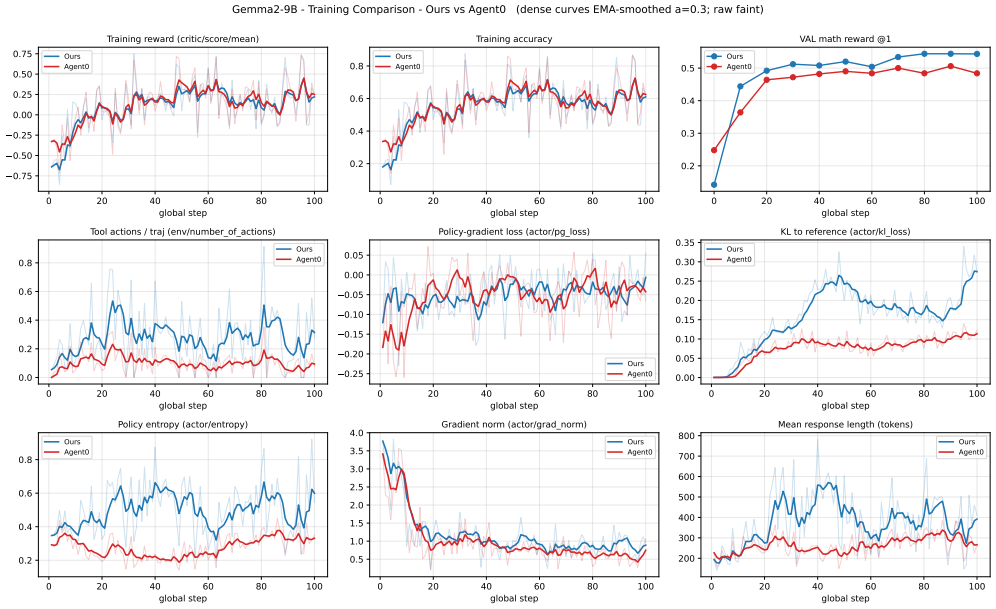

Figure 5: Gemma2-9B based-model

Figure 5 shows nine panels, model length 8,192 as at Gemma3-4B. Training reward and accuracy again move almost together (mean $-0.328 \rightarrow 0.219$ Agent0 vs. $-0.641 \rightarrow 0.250$ Ours; accuracy $0.336 \rightarrow 0.609$ vs. $0.180 \rightarrow 0.625$), with Ours starting from a visibly worse initialization on both — the largest first-step training gap of any backbone in the study — and closing it by roughly step 30. Held-out math validation reward mirrors that pattern and then keeps separating: Ours opens below Agent0 (0.142 vs. 0.248) but crosses over by the second checkpoint and pulls ahead for the rest of training, finishing at 0.544 against 0.484, the smallest final math gain of the four backbones

(+0.49 pp mean row effect, Table 3) but a real and monotonic one. Tool actions per trajectory are higher for Ours throughout, as at every other backbone (plateau ∼0.3 vs. Agent0's ∼0.1–0.15). Two panels invert relative to every other backbone in the study. Policy entropy is *higher* for Ours than for Agent0 for essentially the whole run ($0.347 \rightarrow 0.545$ vs. $0.292 \rightarrow 0.350$, Ours peak 0.922) — at every other backbone Ours's entropy was flat or lower — consistent with a policy exploring a recall action that, for this backbone, never pays off. KL to the reference opens the widest gap of any backbone: the EMA-smoothed curve plotted here peaks near 0.26 by step 45, dips, then climbs back to 0.271 by the final step, against Agent0's 0.123; the raw per-step maxima behind these curves are 0.340 and 0.140 (Table 1). Either way, Ours moves furthest from its reference policy of any run in the study while gaining the least. Gradient norm is the one panel that is nearly identical between arms throughout (both fall from ∼3.5–3.8 at step 1 to ∼0.6–1.0 by the end), unlike the divergent grad-norm dynamics at both Gemma-3-4B and the Qwen pair. Recall usage is non-monotonic, as at Qwen3-8B: it opens at 0.010, peaks at 0.44 around mid-run, and closes at 0.079 ($2{,}881/21{,}899$ rollout actions were recalls overall, 13.16%). Memory-stress reward stays exactly flat at 0 for both arms across the entire run — the one backbone in the study where Ours never converts a single held-out memory-stress item despite issuing recall actions on a rollout share comparable to the other three backbones (11.4–14.5%).