EFFECTIVE-SUPERVISION-AWARE ON-POLICY DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation (OPD) provides token-level teacher supervision on student-generated trajectories. Yet these trajectories differ in answer correctness and in their agreement with teacher predictions, raising a practical question: how should training updates be allocated among them? We study effective-supervision-aware allocation in OPD: using outcome and teacher-based signals to prioritize trajectories for training without changing the distillation loss. As a concrete implementation, we introduce the Reliable Trajectory Buffer (RTB). RTB uses answer correctness and teacher-assigned likelihood to select current trajectories for replacement, teacher–student predictive compatibility to admit previously encountered mistakes to a replay buffer, and rescoring after replacement to refresh distillation signals. Across six mathematical reasoning benchmarks, with Qwen3-8B as teacher, RTB improves the aggregate score from 25.66 to 28.01 for Qwen3-1.7B-Base and from 35.68 to 38.03 for Qwen3-4B-Base. It also reaches higher scores earlier when measured by newly sampled student trajectories. Ablations support the roles of selective replacement and compatibility-based admission.

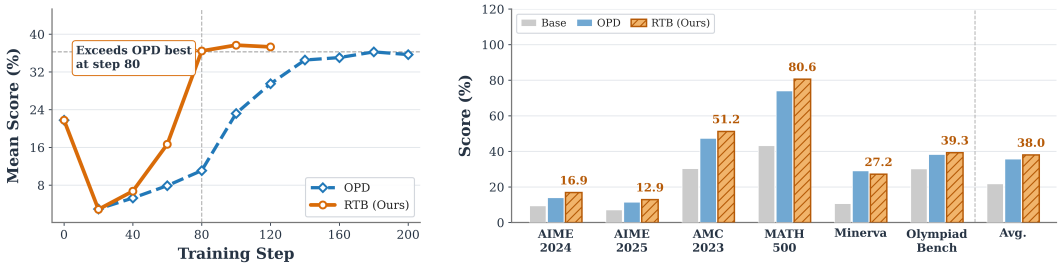


Figure 1: RTB improves sample efficiency and reasoning performance over vanilla OPD with Qwen3-8B as teacher and Qwen3-4B-Base as student. Left: Training dynamics of the aggregate score, computed by averaging Avg@16 on AIME24, AIME25, and AMC23 with Pass@1 on MATH-500, Minerva Math, and OlympiadBench. Right: Performance on the same six benchmarks.

1 INTRODUCTION

On-policy distillation (OPD) trains a student on its own generated trajectories while using a teacher to provide token-level supervision at the states the student visits. This offers supervision on student-encountered prefixes rather than fixed demonstrations, and a denser signal than outcome-only feedback (Agarwal et al., 2024; Gu et al., 2024). Yet student-generated trajectories need not offer equally useful opportunities for teacher-guided learning. This raises an allocation question for OPD: which current trajectories should be optimized, and which previously encountered trajectories should be revisited?

Final-answer correctness is informative but insufficient for this decision. A correct trajectory may already align closely with the teacher, or it may still contain substantial predictive disagreement. Incorrect trajectories can likewise differ in how well teacher and student predictions align on their

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.

visited prefixes. Thus, trajectories with the same outcome may offer different opportunities for distillation. In particular, an incorrect trajectory that remains predictively compatible with the teacher may be worth revisiting: it represents a mistake while still providing states on which teacher-guided supervision is plausible. We call such trajectories *teachable mistakes*. Compatibility is a selection signal, not a guarantee that replaying any particular mistake will help.

We introduce the *Reliable Trajectory Buffer* (RTB), a trajectory-allocation method for OPD. RTB makes three decisions. For newly generated trajectories, it uses answer correctness and teacher-assigned likelihood to select which trajectories to replace rather than optimize as sampled. For previously encountered mistakes, it uses teacher–student predictive compatibility to decide which candidates to admit to a replay buffer. After replacement, RTB rescores the resulting batch under the current student and fixed teacher to refresh the training signals before optimization. In this way, selected historical mistakes receive further teacher supervision in place of selected current trajectories. RTB changes the allocation of training trajectories without changing the token-level distillation loss. Because replayed trajectories were generated by earlier student policies, this allocation deliberately relaxes strict on-policy sampling.

We evaluate RTB on six mathematical reasoning benchmarks, using Qwen3-8B as the teacher and Qwen3-1.7B-Base and Qwen3-4B-Base as students. Our aggregate score is the unweighted mean of Avg@16 on AIME24, AIME25, and AMC23 and Pass@1 on MATH-500, Minerva Math, and OlympiadBench. Compared with vanilla OPD, RTB improves this score from 25.66 to 28.01 for the 1.7B student and from 35.68 to 38.03 for the 4B student. It also reaches higher aggregate scores earlier when training progress is measured by newly sampled student trajectories. Ablations support the roles of both selective replacement and compatibility-based buffer admission. These are gains under a rollout-based comparison and do not, on their own, establish lower total training compute.

Our contributions are:

- We formulate trajectory selection in OPD as an allocation problem in which answer correctness alone does not determine the value of teacher-guided supervision.
- We introduce RTB, which combines selective replacement of current trajectories, compatibility-based admission of historical mistakes, and rescoring under the current student while retaining the original token-level distillation loss.
- We show improvements across six mathematical reasoning benchmarks and two student scales, and assess the roles of replacement and buffer admission through ablations.

2 PRELIMINARIES

On-policy distillation. Let π_θ be a trainable student and π_{T} a fixed teacher sharing a tokenizer and vocabulary $\mathcal{V}$. For a prompt $x \sim \mathcal{D}$, a rollout policy $\pi_{\bar{\theta}}$ generates a response $y = (y_1, \ldots, y_T)$. At a student-generated prefix $s_t = (x, y_{<t})$, let $p_t = \pi_\theta(\cdot \mid s_t)$ and $q_t = \pi_{\mathrm{T}}(\cdot \mid s_t)$ be the student and teacher next-token distributions. OPD trains the student on such visited prefixes using teacher supervision (Agarwal et al., 2024).

For a batch of sampled trajectories $\mathcal{B} = \{(x_i, y_i)\}_{i=1}^{B}$, we use the token-averaged objective

$$\widehat{\mathcal{L}}_{\mathrm{OPD}}(\theta; \mathcal{B}) = \frac{1}{\sum_{i=1}^{B} T_i} \sum_{i=1}^{B} \sum_{t=1}^{T_i} \ell_{i,t}^{\mathrm{KD}}, \tag{1}$$

where T_i is the number of valid response tokens in trajectory i. The rollout policy is refreshed from the student during training, and sampled trajectories are held fixed during an update. Vanilla OPD optimizes on newly sampled trajectories; RTB can additionally optimize on previously sampled trajectories. Thus, replay changes the provenance of some training trajectories, not the per-token loss applied to them.

Distillation loss. We use reverse-KL supervision, following prior work on generative distillation (Gu et al., 2024). For computational efficiency, the loss at prefix s_t is evaluated on the student's top-k_{tr} set

$$\mathcal{K}_t^{\mathrm{tr}} = \mathrm{TopK}(p_t, k_{\mathrm{tr}}). \tag{2}$$

Let $\widetilde{p}_t$ and $\widetilde{q}_t$ denote the student and teacher distributions restricted to this shared token set and renormalized within it. The token-level loss is

$$\ell_t^{\mathrm{KD}} = \sum_{v \in \mathcal{K}_t^{\mathrm{tr}}} \widetilde{p}_t(v) \log \frac{\widetilde{p}_t(v)}{\widetilde{q}_t(v)}. \tag{3}$$

Because probability mass outside $\mathcal{K}_t^{\mathrm{tr}}$ is excluded, this restricted-support loss is not identical to full-vocabulary $D_{\mathrm{KL}}(p_t \| q_t)$. Vanilla OPD and RTB use the same loss; they differ in which trajectories are included in the training batch.

Predictive compatibility. To characterize teacher–student agreement on a visited prefix, we compare their high-probability token sets. Let $\mathcal{K}_t^{\mathrm{S}} = \mathrm{TopK}(p_t, k_{\mathrm{diag}})$ and $\mathcal{K}_t^{\mathrm{T}} = \mathrm{TopK}(q_t, k_{\mathrm{diag}})$. Their top-$k$ overlap is

$$o_t = \frac{|\mathcal{K}_t^{\mathrm{S}} \cap \mathcal{K}_t^{\mathrm{T}}|}{k_{\mathrm{diag}}} \in [0, 1]. \tag{4}$$

We set $k_{\mathrm{diag}} = 16$ for compatibility measurement and $k_{\mathrm{tr}} = 16$ for the distillation loss. Although the values coincide, they serve different purposes. Overlap describes local predictive agreement, not answer correctness or the expected gain from training on a trajectory. RTB combines this diagnostic signal with outcome information when selecting erroneous trajectories for replay, as described in Section 4.

3 Related Work

Distillation objectives for language models. Knowledge distillation transfers teacher capabilities to smaller language models through distribution matching. GKD (Agarwal et al., 2024) studies distillation on student-generated contexts with flexible divergence objectives, mitigating the mismatch between training and inference distributions. MiniLLM (Gu et al., 2024) develops reverse-KL distillation with policy-gradient optimization. DistiLLM-2 (Ko et al., 2025) introduces a contrastive formulation that increases the likelihood of teacher responses while decreasing that of student responses. These methods develop objectives and optimization procedures for transferring teacher knowledge.

Data selection and supervision allocation in distillation. Beyond the distillation objective, recent work examines which student-generated contexts provide useful teacher supervision. Li et al. (2026a) connects successful OPD to compatible thinking patterns and knowledge that the teacher can contribute beyond the student's existing capabilities. At the token and prefix levels, Prefix OPD (Zhang et al., 2026) restricts distillation to initial response segments, while TIP (Xu et al., 2026) selects informative positions using student entropy and teacher–student divergence. Entropy-Aware OPD (Jin et al., 2026) adjusts the distillation objective at positions with high teacher uncertainty. Prune-OPD (Yang et al., 2026) uses predictive compatibility to detect drift and adapt supervision and rollout length. At the trajectory level, SCOPE (**?**) routes correct rollouts to student-perplexity-weighted MLE and incorrect rollouts to teacher-perplexity-weighted distillation. RA-OPD (**?**) filters trajectories whose distillation returns conflict with their outcome rewards. FiRe-OPD (Li et al., 2026b) combines trajectory filtering with token reweighting, while DistiLLM (Ko et al., 2024) reuses historical student outputs through adaptive off-policy distillation. RTB instead couples selective replacement of current rollouts with compatibility-based replay of historical mistakes. It uses correctness and teacher likelihood to identify replacement targets, admits mistakes based on predictive compatibility, and rescores replay candidates under the current student, while retaining the baseline token-level distillation loss.

4 Method

4.1 Motivation

Dense teacher supervision does not make every student-generated trajectory equally useful for distillation. Some successful responses may offer limited room for further task-level improvement, while some unsuccessful responses lead to prefixes on which teacher guidance provides little corrective

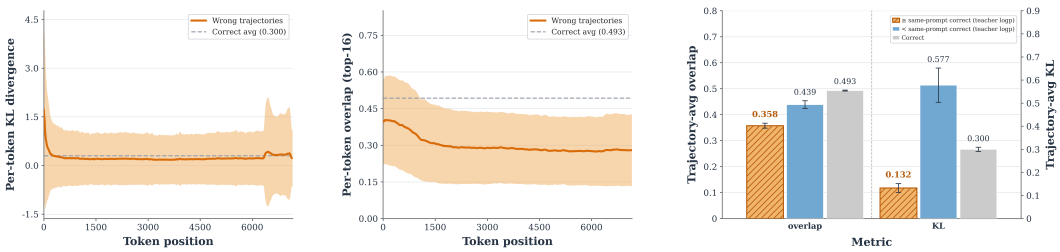


Figure 2: Teacher–student diagnostics on AMC23 with Qwen3-1.7B-Base as student and Qwen3-8B as teacher. Left/middle: Token-level KL and top-16 overlap on incorrect trajectories, with correct-response averages as dashed references. Right: Trajectory-level comparison of correct responses and incorrect responses grouped by whether their average teacher log-likelihood exceeds the correct-response mean for the same prompt.

signal. If all trajectories are treated uniformly, training may spend updates on responses with limited remaining correction need and on errors that provide weak local guidance, leaving fewer updates for more promising learning opportunities. This motivates selecting trajectories according to both their remaining learning need and the usefulness of teacher supervision.

Reallocating updates from selected correct trajectories. A correct response has achieved the task objective, but correctness alone does not imply that further distillation is unnecessary. We therefore also consider the teacher's length-normalized likelihood of the generated response. Correct responses with high teacher likelihood combine a successful outcome with strong teacher support for their observed content. We use this combination as a heuristic for reallocating some of their updates toward unresolved errors. Only a fraction of these responses is replaced; the others remain in the optimization batch to preserve learning opportunities that may remain despite a correct final answer.

Identifying errors with limited corrective guidance. Incorrect responses require correction, but their prefixes may not always provide useful teacher guidance. Figure 2 shows that incorrect AMC23 trajectories have lower mean top-16 overlap than correct responses, while their token-level KL falls below the correct-response reference through most later positions. To identify errors exhibiting this pattern, we compare each incorrect response's length-normalized teacher log-likelihood with the mean for correct responses to the same prompt. Errors above this reference, termed *teacher confidence mismatch*, have lower overlap (0.358 vs. 0.439) and lower KL (0.132 vs. 0.577) than errors below it. Their KL is also below the correct-response average (0.300). This pattern motivates replacing a subset of these errors: strong teacher support for an incorrect response can coincide with weak predictive compatibility and a small local distillation discrepancy.

Preserving teachable mistakes for future updates. Among incorrect responses, some retain prefixes on which teacher and student predictions remain compatible. Their incorrect outcomes indicate an unresolved task-level need, while higher top-k overlap suggests that teacher guidance may still be useful on these contexts. We refer to such cases as *teachable mistakes*. Storing their prefixes in a buffer preserves these potential learning opportunities beyond the batch in which they arise, making them available for future distillation updates.

4.2 Reliable Trajectory Buffer

Figure 3 gives an overview of RTB. At each training step, RTB proceeds in three steps: selecting current rollouts for replacement, selecting incorrect rollouts for buffer admission, and refreshing distillation signals after replacement. Selected current rollouts are replaced with trajectories sampled from the existing buffer, and the resulting batch is rescored under the current student and fixed teacher. Newly selected admission candidates become available for replay from the next training step. The token-level distillation loss remains unchanged.

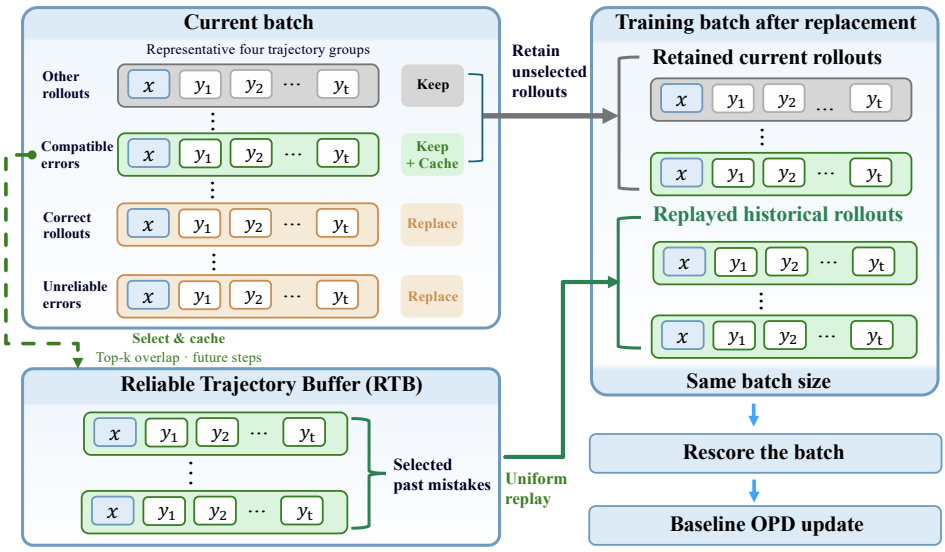


Figure 3: Overview of RTB. Selected rollouts (red) are replaced with buffered mistakes, while compatible errors (green) are retained for future replay.

At training step s, the student rollout policy $\pi_{\tilde{\theta}_s}$ generates $\mathcal{B}_s = \{\tau_i\}_{i=1}^{B}$, where $\tau_i = (x_i, y_i)$ denotes a prompt–response trajectory. Let $c_i \in \{0, 1\}$ denote its outcome correctness under the task verifier.

Selecting replacement targets. RTB uses outcome correctness to distinguish successful and unsuccessful rollouts, and teacher likelihood to select replacement targets within each category. For a trajectory $\tau = (x, y)$, we measure teacher support by the average response log-likelihood:

$$\lambda_T(\tau) = \frac{1}{T} \sum_{t=1}^{T} \log \pi_T(y_t \mid x, y_{<t}), \tag{5}$$

where T counts valid response tokens. This score measures teacher support for the observed response, rather than agreement between the full teacher and student distributions.

Among correct rollouts, RTB selects the α fraction with the highest λ_T for replacement. These responses combine a successful outcome with strong teacher support, motivating the reallocation of some of their updates to unresolved errors.

For incorrect rollouts, teacher support is compared with correct responses to the same prompt. Let

$$\mathcal{C}_s(x) = \{i : x_i = x,\ c_i = 1\}.$$

When $\mathcal{C}_s(x) \neq \varnothing$, define

$$b_T(x) = \frac{1}{|\mathcal{C}_s(x)|} \sum_{i \in \mathcal{C}_s(x)} \lambda_T(\tau_i). \tag{6}$$

The incorrect replacement candidates are

$$\mathcal{I}_{\mathrm{w}} = \{i : c_i = 0,\ |\mathcal{C}_s(x_i)| > 0,\ \lambda_T(\tau_i) > b_T(x_i)\}\,. \tag{7}$$

These candidates exhibit teacher confidence mismatch: an incorrect response receives greater teacher support than the correct-response mean. RTB selects the α fraction of these candidates with the highest λ_T for replacement. We use $\alpha = 0.5$ for both selections.

All unselected rollouts remain in the current batch. In particular, all-wrong prompt groups lack a correct-response reference and are not selected by this replacement rule. Their incorrect rollouts remain eligible for buffer admission.

Method	AIME24		AIME25		AMC23		MATH500	MINERVA	OLYMPIAD	Average	
	Avg@16	Pass@16	Avg@16	Pass@16	Avg@16	Pass@16	Pass@1	Pass@1	Pass@1	Avg@n	Pass@n
Qwen3-8B	29.38	60.00	23.33	43.33	69.84	92.50	83.40	27.94	49.41	47.22	59.43
Student Backbone: Qwen3-1.7B-Base											
Qwen3-1.7B-Base	2.29	20.00	1.67	10.00	20.16	62.50	30.20	9.56	12.61	12.75	24.15
OPD	8.33	**23.33**	4.58	**23.33**	33.44	72.50	64.00	16.91	26.71	25.66	37.80
RTB (Ours)	**8.96**	20.00	**5.21**	**23.33**	**37.81**	**75.00**	**66.80**	**20.96**	**28.34**	**28.01**	**39.07**
Δ vs. OPD	+7.6%	−14.3%	+13.8%	0.0%	+13.1%	+3.4%	+4.4%	+24.0%	+6.1%	+9.2%	+3.4%
Student Backbone: Qwen3-4B-Base											
Qwen3-4B-Base	9.38	30.00	7.08	30.00	30.31	77.50	43.20	10.66	30.12	21.79	36.91
OPD	13.96	23.33	11.46	**40.00**	47.34	80.00	74.00	**29.04**	38.28	35.68	47.44
RTB (Ours)	**16.88**	**36.67**	**12.92**	**40.00**	**51.25**	**82.50**	**80.60**	27.21	**39.32**	**38.03**	**51.05**
Δ vs. OPD	+20.9%	+57.2%	+12.7%	0.0%	+8.3%	+3.1%	+8.9%	−6.3%	+2.7%	+6.6%	+7.6%

Table 1: Comparison of the teacher, base students, vanilla OPD, and our buffer-enhanced OPD. Best results within each student backbone are shown in **bold**.

Admitting teachable mistakes. Admission selects errors to preserve for future distillation using predictive compatibility. Whereas λ_T scores the generated tokens, top-k overlap compares the teacher's and student's likely next-token candidates on the contexts to be stored.

Let $P_s(\tau)$ denote the retained response prefix, and let $\mathcal{I}(\tau)$ denote the valid response positions of a trajectory or prefix. Using Eq. equation 4, define

$$\kappa_s(\tau) = \frac{1}{|\mathcal{I}(P_s(\tau))|} \sum_{t \in \mathcal{I}(P_s(\tau))} o_t. \tag{8}$$

RTB considers all incorrect rollouts in the fresh batch, including those from mixed and all-wrong prompt groups, and selects up to M with the highest κ_s for admission. Correctness is evaluated on the complete response before truncation, while compatibility is measured on the retained prefix at collection time.

Admission and replacement are independent selections. An incorrect rollout can exhibit confidence mismatch on its complete response while having a compatible prefix selected for admission. The selected prompt–prefix pairs are copied into a pending admission set before replacement; this does not itself remove the original rollouts from the current batch.

Replacing and replaying trajectories. For each selected replacement target, RTB samples an entry uniformly with replacement from the existing buffer $\mathcal{H}_s$. The sampled prompt–prefix pair replaces the selected rollout, preserving the replayed entry's original prompt. Let $\widetilde{\mathcal{B}}_s = \{\widetilde{\tau}_i\}_{i=1}^{B}$ denote the resulting optimization batch. If the buffer is empty, the batch remains unchanged.

Pending admissions are appended only after replay and rescoring, making them available from the next training step. The buffer stores at most C entries and evicts the oldest entries when this capacity is exceeded.

Refreshing signals and optimizing the batch. Whenever at least one replacement succeeds, RTB rescores the entire resulting batch $\widetilde{\mathcal{B}}_s$ under the current student and fixed teacher to refresh the training signals before optimization. This evaluates stored responses using the current models; it does not regenerate the responses or make historical trajectories on-policy.

We then apply the same token-level distillation loss as vanilla OPD:

$$\widehat{\mathcal{L}}_{\mathrm{RTB}}(\theta) = \frac{1}{Z_s} \sum_{i=1}^{B} \sum_{t \in \mathcal{I}(\widetilde{\tau}_i)} \ell_t^{\mathrm{KD}}(\theta; \widetilde{\tau}_i), \qquad Z_s = \sum_{i=1}^{B} |\mathcal{I}(\widetilde{\tau}_i)|. \tag{9}$$

Here, ℓ_t^{KD} is defined in Eq. equation 3. The student distribution is evaluated under the trainable parameters θ, while response tokens and discrete selection decisions are held fixed during optimization. Outcome correctness is used only for trajectory selection, not as an additional reward. RTB therefore changes the trajectories used for training while preserving the baseline token-level objective.

Algorithm 1 Reliable Trajectory Buffer (RTB)

Given: student π_θ, teacher π_T, dataset $\mathcal{D}$, verifier c
Parameters: buffer capacity C, admission budget M, replacement fraction $\alpha = 0.5$, prefix policy P_s
Initialize FIFO buffer $\mathcal{H} \leftarrow \varnothing$
for $s = 1, \ldots, S$ **do**
    Generate batch $\mathcal{B}_s$; compute correctness c_i and teacher likelihood $\lambda_T(\tau_i)$
    Select the top-α correct rollouts and incorrect candidates from Eq. equation 7 by λ_T as replacement set $\mathcal{J}_s$
    Select up to M incorrect rollouts with highest prefix compatibility κ_s; copy them to pending set $\mathcal{A}_s$
    Set $\widetilde{\mathcal{B}}_s \leftarrow \mathcal{B}_s$; replace targets in $\mathcal{J}_s$ by uniform replay from $\mathcal{H}$ when nonempty
    if at least one replacement succeeds **then**
        Rescore $\widetilde{\mathcal{B}}_s$ under the current student and fixed teacher
    end if
    Append $\mathcal{A}_s$ to $\mathcal{H}$ and evict oldest entries beyond capacity C
    Update θ using Eq. equation 9
end for

4.3 Implementation Details

We use a prompt-group-dependent length cap to construct the retained prefix $P_s(\tau)$. The cap is linearly increased during training, while responses shorter than the cap are retained in full. The schedule is provided in Appendix A.3.

The buffer stores prompts, retained response prefixes, and task metadata, excluding policy-dependent training scores. Replayed entries are aligned to the current batch format while preserving their original prompts. If an entry cannot be aligned within the allowed sampling attempts, the original rollout is retained. Algorithm 1 summarizes the procedure; buffer settings and alignment details are provided in Appendix A.2.

5 Experiments

5.1 Experimental Setup

Training Setup. We train on DAPO-Math-17K (Yu et al., 2025), which contains approximately 17K mathematical reasoning problems with verifiable answers. We use Qwen3-8B (Yang et al., 2025) as the teacher and Qwen3-1.7B-Base and Qwen3-4B-Base as students. We compare our method with vanilla OPD using the same training prompts, teacher–student pairs, and optimization settings, and report the original student and teacher models as references. Detailed training hyperparameters are provided in Appendix A.1.

Evaluation. We evaluate on six mathematical reasoning benchmarks: AIME24, AIME25, AMC23, MATH-500 (Hendrycks et al., 2021; Lightman et al., 2024), Minerva Math (Lewkowycz et al., 2022), and OlympiadBench (He et al., 2024). We report Avg@16 and Pass@16 on AIME24, AIME25, and AMC23, and Pass@1 on the remaining benchmarks. For aggregate results, we macro-average the corresponding scores across all six benchmarks.

5.2 Main Results

Overall Performance. Table 1 compares RTB with vanilla OPD across two student scales and six reasoning benchmarks. Vanilla OPD already improves substantially over the base models, while RTB consistently provides further gains. For Qwen3-1.7B-Base, RTB improves aggregate Avg@n from 25.66 to 28.01 and Pass@n from 37.80 to 39.07. For Qwen3-4B-Base, the corresponding scores increase from 35.68 to 38.03 and from 47.44 to 51.05, respectively.

Method	AIME24	AIME25	AMC23	MATH-500	MINERVA	OLYMPIAD-BENCH	Average
Student Backbone: Qwen3-1.7B-Base							
OPD	8.33	4.58	33.44	64.00	16.91	26.71	25.66
w/o Error Replacement	8.75	5.42	35.47	65.60	**20.96**	26.41	26.93
w/o Correct Replacement	8.13	**6.46**	**40.31**	65.00	17.65	**29.38**	27.82
RTB (Ours)	**8.96**	5.21	37.81	**66.80**	**20.96**	28.34	**28.01**

Table 2: Ablation on different replacement targets. For AIME24, AIME25, and AMC23, we report Avg@16; for the remaining benchmarks, we report Pass@1.

Method	AIME24	AIME25	AMC23	MATH-500	MINERVA	OLYMPIAD-BENCH	Average
Student Backbone: Qwen3-1.7B-Base							
RTB (Random)	8.75	3.75	37.34	64.00	17.65	**30.12**	26.93
RTB (Overlap)	**8.96**	**5.21**	**37.81**	**66.80**	**20.96**	28.34	**28.01**

Table 3: Ablation on buffer trajectory selection. Random admission is compared with the proposed top-k overlap-based selection.

The improvement is also broadly distributed across individual benchmarks rather than being driven by a single task. Across the 18 benchmark-specific metrics for the two student models, RTB improves 14 and matches two relative to vanilla OPD. The only decreases occur on AIME24 Pass@16 for the 1.7B student and Minerva Math Pass@1 for the 4B student. These results demonstrate that RTB consistently strengthens vanilla OPD across different model scales and reasoning benchmarks.

5.3 Ablation Studies

We conduct ablation studies on Qwen3-1.7B-Base to examine two key design choices: which low-value trajectories should be replaced and which trajectories should be admitted into the buffer.

Which trajectories should be replaced? RTB targets two types of potentially low-value trajectories: teacher-supported correct responses and incorrect responses exhibiting teacher confidence mismatch. We disable replacement of each type separately, yielding *w/o Correct Replacement* and *w/o Error Replacement*, respectively. Table 2 shows that the full method achieves the highest aggregate score of 28.01, compared with 27.82 without correct-response replacement and 26.93 without error-response replacement. The larger drop from disabling error replacement suggests that reallocating updates away from errors with teacher confidence mismatch is particularly beneficial in this setting. Correct-response replacement provides a smaller additional gain, supporting the combined strategy.

Does overlap-based selection matter? To isolate the effect of predictive compatibility, we replace overlap-based buffer admission with random selection among incorrect trajectories, keeping the remaining procedure unchanged. As shown in Table 3, overlap-based admission yields broadly distributed gains and improves the aggregate score from 26.93 to 28.01. This suggests that the choice of replayed mistakes matters beyond the use of error replay alone, supporting predictive compatibility as a useful criterion for buffer admission. Together, the ablations support both components of RTB: selecting which trajectories to replace and which mistakes to retain for reuse.

5.4 Trajectory Selection with Pseudo-Labels

To assess RTB's reliance on ground-truth answers, we replace them with teacher-generated references for trajectory selection. Using Qwen3-1.7B-Base as the student, we obtain a reference answer from Qwen3-8B for each training problem without checking it against ground truth. Agreement with this reference replaces the correctness indicator used in buffer admission and trajectory replacement; the remaining procedure is unchanged.

Table 4 shows that pseudo-label selection improves the average score from 25.66 for vanilla OPD to 27.62, compared with 28.01 using ground-truth answers. It retains approximately 83% of the

Method	AIME24	AIME25	AMC23	MATH-500	MINERVA	OLYMPIAD-BENCH	Average
OPD	8.33	4.58	33.44	64.00	16.91	26.71	25.66
RTB (Pseudo)	**9.79**	**6.46**	37.19	**66.80**	16.54	**28.93**	27.62
RTB (GT)	8.96	5.21	**37.81**	**66.80**	**20.96**	28.34	**28.01**

Table 4: Trajectory selection using ground-truth answers (GT) or teacher-generated reference answers (Pseudo). We report Avg@16 for AIME24, AIME25, and AMC23, and Pass@1 for the remaining benchmarks. Best results are shown in **bold**.

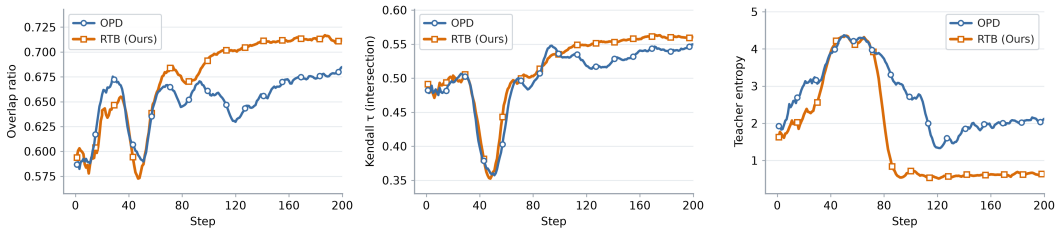


Figure 4: Predictive compatibility and teacher uncertainty on training batches. Left: Top-k overlap. Middle: Kendall's τ over shared top-k tokens. Right: Teacher predictive entropy.

aggregate improvement and outperforms vanilla OPD on five of the six benchmarks. These results suggest that teacher-generated references can provide a practical alternative for RTB's trajectory selection when ground-truth training answers are unavailable.

5.5 Predictive Compatibility and Teacher Uncertainty

Figure 4 examines whether RTB changes the predictive characteristics of the trajectories used for optimization. We measure compatibility through top-k overlap and Kendall's τ between teacher and student rankings within their shared top-k tokens. These metrics capture complementary aspects of agreement: candidate membership and relative preference. After early fluctuations, RTB generally maintains higher overlap than vanilla OPD through the middle and later training stages, accompanied by stronger ranking agreement. Thus, its selected training contexts exhibit not only more shared candidate tokens, but also greater consistency in how the two models rank them.

Teacher predictive entropy decreases earlier under RTB and remains lower in later training stages, indicating reduced teacher uncertainty on the selected contexts. Together with higher token overlap and stronger ranking agreement, this suggests that RTB shifts the training distribution toward trajectories where teacher predictions are more concentrated and better aligned with the student's candidate tokens and relative preferences. These trends support RTB's intended role of prioritizing student mistakes that remain compatible with teacher guidance.

6 Conclusion

In this work, we study the problem of trajectory quality selection in on-policy distillation. We show that uniformly applying teacher supervision to all student-generated rollouts can lead to inefficient updates, as correct trajectories may require limited additional task-level correction, while some erroneous trajectories exhibit weak predictive compatibility and limited local corrective signal. We propose Reliable Trajectory Buffer (RTB), a trajectory replacement framework that redirects distillation toward error trajectories that remain informative and compatible with teacher supervision. RTB preserves the original token-level OPD objective while adapting the trajectory distribution used for optimization. Experiments across two student scales and six mathematical reasoning benchmarks demonstrate that RTB consistently improves over vanilla OPD and achieves better sample efficiency under matched training conditions. Our findings suggest that improving OPD requires not only better distillation objectives, but also supervision-aware selection of trajectories where teacher guidance can be most effectively utilized.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, 2024.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *International Conference on Learning Representations*, 2024.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 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. Association for Computational Linguistics, 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. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021.

Woogyeol Jin, Taywon Min, Yongjin Yang, Swanand Ravindra Kadhe, Yi Zhou, Dennis Wei, Nathalie Baracaldo, and Kimin Lee. Entropy-Aware on-policy distillation of language models. *arXiv preprint arXiv:2603.07079*, 2026.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. DistiLLM: Towards streamlined distillation for large language models. In *International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 24872–24895. PMLR, 2024.

Jongwoo Ko, Tianyi Chen, Sungnyun Kim, Tianyu Ding, Luming Liang, Ilya Zharkov, and Se-Young Yun. DistiLLM-2: A contrastive approach boosts the distillation of LLMs. In *International Conference on Machine Learning*, volume 267, pp. 31044–31062, 2025.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 3843–3857, 2022.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huan ang Gao, Wenkai Yang, Zhiyuan Liu, and Ning Ding. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026a.

Yuying Li, Leqi Zheng, Yongzi Yu, Wenrui Zhou, Xuchang Zhong, Xing Hu, Jing Jin, Hangjie Yuan, and Tao Feng. Filter, then reweight: Rethinking optimization granularity in on-policy distillation. *arXiv preprint arXiv:2606.02684*, 2026b.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri 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*, 2024.

Yuanda Xu, Hejian Sang, Zhengze Zhou, Ran He, Zhipeng Wang, and Alborz Geramifard. TIP: Token importance in on-policy distillation. *arXiv preprint arXiv:2604.14084*, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Zhicheng Yang, Zhijiang Guo, Yifan Song, Minrui Xu, Yongxin Wang, Yiwei Wang, Xiaodan Liang, and Jing Tang. Prune-OPD: Efficient and reliable on-policy distillation for long-horizon reasoning. *arXiv preprint arXiv:2605.07804*, 2026.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Dongxu Zhang, Zhichao Yang, Sepehr Janghorbani, Jun Han, Andrew Ressler, II, Qian Qian, Gregory D. Lyng, Sanjit Singh Batra, and Robert E. Tillman. Fast and effective on-policy distillation from reasoning prefixes. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 25553–25569. Association for Computational Linguistics, 2026.

A APPENDIX

A.1 TRAINING HYPERPARAMETERS

We summarize the training hyperparameters used throughout our experiments in Table 5. Unless otherwise specified, we use the same optimization, generation, and distillation configurations for all OPD-based methods to ensure a controlled comparison. For each prompt, the student samples four on-policy responses, and the teacher provides token-level supervision on the student-generated trajectories. The trajectory buffer is disabled for the vanilla OPD baseline and enabled only for our method. We use the same decoding configuration across all evaluations.

Table 5: Training hyperparameters used in our OPD experiments. Settings are shared across methods unless otherwise specified.

Hyperparameter	**Value**
Optimization	
Learning rate	1×10^{-6}
Training batch size	64 prompts
Mini-batch size	64 prompts
Generation	
Maximum prompt length	1,024
Maximum response length	7,168
Responses per prompt (G)	4
Sampling temperature	1.0
On-Policy Distillation	
Teacher temperature	1.0
Log-probability top-k	16
Compatibility overlap top-k	16
Top-k support	Student
Token weighting	Student probability
Loss aggregation	Token mean
Token-level supervision	Direct teacher reward
Method-Specific	
Trajectory buffer (OPD)	Disabled
Trajectory buffer (Ours)	Enabled
Format reward	Disabled
Evaluation	
Validation temperature	0.7
Validation top-p	0.95
Model data type	FP32

A.2 RTB IMPLEMENTATION DETAILS

Replacement and admission settings. We set the replacement fraction to $\alpha = 0.5$ for both correct and incorrect rollouts. Among correct rollouts, we select the highest-teacher-likelihood α fraction for replacement. For incorrect rollouts, we first construct the candidate set in Eq. equation 7 and then select its highest-teacher-likelihood α fraction. The additional replacement-ratio multiplier is set to 1.0 and does not further reduce these selections. Admission considers all incorrect rollouts and retains up to $M = 16$ candidates per step, ranked by predictive compatibility on their retained prefixes. No additional student–teacher likelihood comparison is imposed on admission.

Buffer storage and sampling. The buffer is a FIFO queue with capacity $C = 256$ individual rollouts. Each entry stores a deep copy of the original prompt, the retained response prefix, the associated masks and positional information, and task metadata. Policy-dependent quantities, including log-probabilities, rewards, and advantages, are not stored. Each replacement is sampled independently and uniformly with replacement; sampling does not remove an entry. Consequently, an entry may be replayed more than once within the same training batch.

Delayed insertion. Admission candidates are copied from the fresh batch before any replacement occurs. They remain pending while replacements are sampled from the existing buffer and the resulting batch is rescored. Only then are the pending entries appended, with the oldest entries evicted as necessary. Thus, a trajectory admitted at step s cannot be replayed before step $s + 1$.

Batch alignment. A replayed entry retains its original prompt and task metadata. Prompt tokens are left-padded and response tokens are right-padded to the current batch dimensions, with masks and positional information adjusted accordingly. If an entry's effective prompt length exceeds the available prompt dimension, another entry is sampled. We allow up to eight alignment attempts per target; if all attempts fail, the original rollout is retained. Response content exceeding the available response dimension is truncated.

Full-batch rescoring. Whenever at least one replacement succeeds, we discard the existing policy-dependent scores and rerun student and teacher scoring over the entire resulting batch. We then recompute distillation rewards and the corresponding training signals. Full-batch rescoring accommodates the data-parallel partitioning requirements of the training framework and avoids combining stale scores with newly computed ones. The update uses direct token-level rewards without within-prompt advantage normalization. Correctness scores serve trajectory selection and do not provide an additional outcome-reward term.

A.3 Auxiliary Prefix Truncation

Group-dependent length caps. Prefix truncation is applied only to incorrect rollouts considered for buffer admission. A prompt group is *all-wrong* if none of its sampled responses is correct, and *mixed* if it contains both correct and incorrect responses. Let $g(\tau) \in \{\mathrm{aw}, \mathrm{mix}\}$ denote the group type of an incorrect rollout. We assign initial retention ratios $r^{(0)}_{\mathrm{aw}} = 0.5$ and $r^{(0)}_{\mathrm{mix}} = 0.75$.

Let $L_{\max}$ be the configured maximum response length and s_{ann} the annealing endpoint. The retention ratio at step s is

$$r_g(s) = r^{(0)}_g + \left(1 - r^{(0)}_g\right) \min\left(\frac{s}{s_{\mathrm{ann}}}, 1\right). \tag{10}$$

For a response of length T, the retained length is

$$K_s(\tau) = \min\left\{T,\ \left\lfloor L_{\max} r_{g(\tau)}(s) \right\rfloor\right\}, \qquad P_s(\tau) = \left(x, y_{1:K_s(\tau)}\right). \tag{11}$$

We use $L_{\max} = 8192$ and $s_{\mathrm{ann}} = 200$. The ratios determine caps relative to the configured maximum length, rather than fractions of each response's own length. Responses shorter than the cap remain unchanged.

Interaction with admission and replay. Correctness is evaluated on the complete response before truncation. The original outcome determines whether the trajectory is an admission candidate; the retained prefix is not independently required to contain a final answer. Compatibility is averaged over the retained positions using Eq. equation 8, and the highest-scoring candidates are admitted. Only the retained prefixes are stored.

The schedule applies to newly considered candidates. Previously stored prefixes are not extended as the length cap increases. Once the schedule reaches full retention, newly admitted responses are stored without this auxiliary truncation, while shorter historical entries may remain until evicted. This procedure imposes a length cap and does not explicitly detect a reasoning error or a degraded suffix.

A.4 Training Dynamics and Sample Efficiency

Figure 5 complements the aggregate results in Figure 1 by showing benchmark-level training dynamics across all six tasks and both student scales.

Learning Dynamics. RTB's improvements are visible before the final checkpoint and extend across both student scales. For Qwen3-1.7B-Base, RTB separates from vanilla OPD during the intermediate stage on AIME24, AIME25, and AMC23, while exhibiting fluctuations at later steps. On

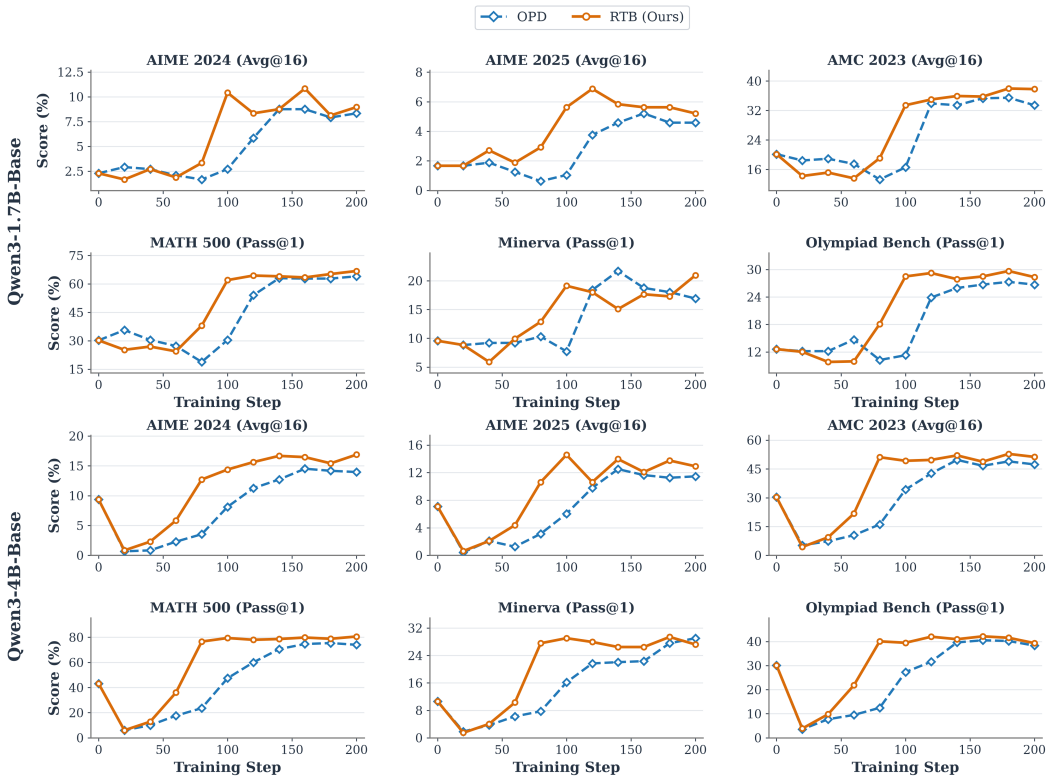


Figure 5: Training dynamics across six reasoning benchmarks. All experiments use Qwen3-8B as the teacher. The upper two rows correspond to Qwen3-1.7B-Base, and the lower two rows to Qwen3-4B-Base. For each student, we report Avg@16 on AIME24, AIME25, and AMC23, and Pass@1 on MATH-500, Minerva Math, and OlympiadBench. Each marker denotes an evaluated checkpoint. RTB often reaches stronger performance at earlier training steps, although the magnitude and persistence of its advantage vary across benchmarks.

MATH-500, Minerva Math, and OlympiadBench, RTB initially trails or remains close to OPD, then achieves higher scores at later evaluated checkpoints. Thus, its advantage is not uniform throughout training, but emerges at different stages across tasks.

For Qwen3-4B-Base, both methods exhibit a pronounced initial performance drop. RTB subsequently recovers earlier, with particularly clear intermediate gains on AMC23, MATH-500, Minerva Math, and OlympiadBench. On several of these tasks, RTB reaches a high-performing plateau while vanilla OPD is still improving. The gap narrows later, and RTB does not dominate every checkpoint: on Minerva Math, for example, vanilla OPD slightly exceeds RTB at the final evaluated step. These trajectories support faster attainment of strong performance rather than uniformly smoother or consistently superior training dynamics.

Interpretation of Sample Efficiency. Both methods use the same rollout batch size and baseline token-level distillation loss. Matched training steps therefore correspond to comparable numbers of newly sampled student trajectories. The earlier gains of RTB, particularly for the 4B student, provide evidence that selective replay improves learning progress relative to this fresh-rollout budget. The benchmark-level results complement the aggregate comparison in Figure 1, while showing that the benefit varies across tasks and training stages.

This notion of sample efficiency concerns newly generated trajectories, rather than matched token counts or computational cost. Response lengths can differ, and replay reuses historical data while

replacement triggers additional scoring. The curves therefore do not establish reductions in wall-clock time, GPU hours, or total teacher-scored tokens.