MULTI-TURN ON-POLICY DISTILLATION WITH PREFIX REPLAY

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-turn on-policy distillation (OPD) gives LLM agents dense teacher supervision on student-generated actions, but requires costly environment rollouts at every update. We propose Replayed-Prefix On-Policy Distillation (ReOPD), which reuses recorded teacher trajectories: the student acts at selected teacher-forced prefixes and receives token-level teacher supervision without interacting with the environment. We identify a prefix trap: student-generated histories improve relevance to the student but can make teacher targets unreliable, while errors compound across turns. A two-sided bound captures the trade-off between student occupancy mismatch and teacher reliability. ReOPD uses a step-decaying sampling schedule that favors earlier, lower-shift prefixes. Across Python-assisted mathematical reasoning and search tasks, with multiple teacher and student scales, ReOPD matches or improves online OPD accuracy, uses zero tool calls during student training, and is at least 4× faster per rollout. Reusing teacher trajectories also supports joint distillation across environments.

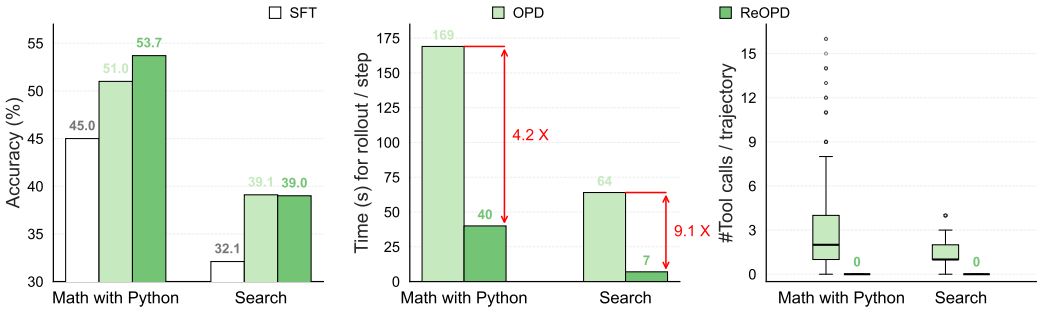


Figure 1: **ReOPD keeps the benefits of on-policy distillation while removing environment interaction.** ReOPD matches or improves OPD accuracy, but trains much faster per step and eliminates tool calls during training by replaying teacher-recorded prefixes instead of executing fresh environment rollouts. Student: Qwen3-4B-Instruct-2507; teacher: Qwen3-8B.

1 INTRODUCTION

Reinforcement learning (RL) has advanced LLM reasoning and tool use through methods such as PPO (Schulman et al., 2017) and GRPO (Shao et al., 2024) with verifiable rewards (DeepSeek-AI et al., 2025; Yu et al., 2025). Its on-policy rollouts expose models to their own mistakes, but scalar rewards provide sparse supervision (Lu & Thinking Machines Lab, 2025). Knowledge distillation instead offers dense teacher supervision (Hinton et al., 2015; Kim & Rush, 2016). Standard off-policy distillation uses teacher-generated prefixes; student mistakes at test time can lead to unseen prefixes and compounding errors (Ross et al., 2011; Ranzato et al., 2016). *On-policy distillation* (OPD) addresses this by supervising student-sampled trajectories with the teacher's per-token distribution (Gu et al., 2024; Agarwal et al., 2024; Lu & Thinking Machines Lab, 2025).

Most OPD work considers single-turn generation, whereas agents alternate actions such as code execution and retrieval with environment feedback over multiple turns (Yao et al., 2022; Schick

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.

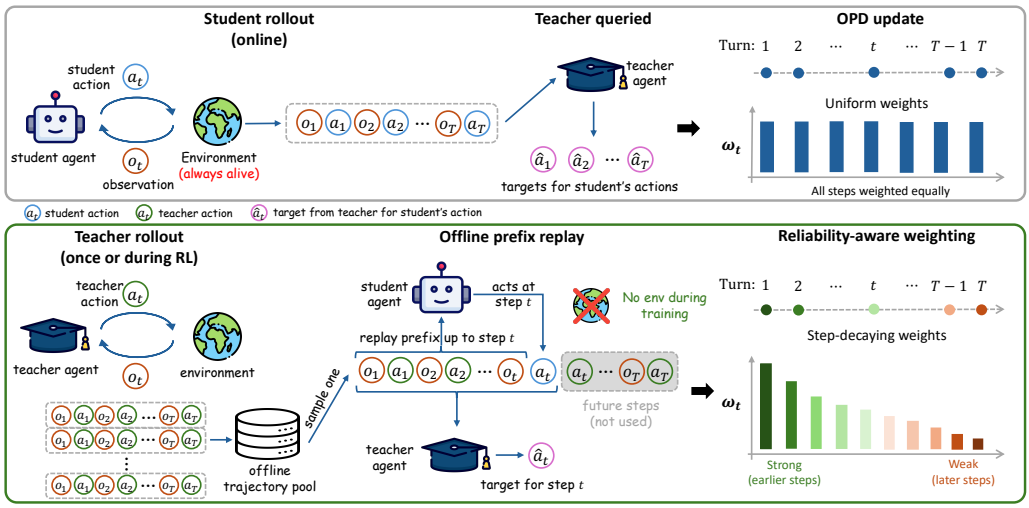


Figure 2: **Online OPD vs ReOPD.** OPD rolls the student through a live environment and supervises every step. ReOPD replays teacher-collected prefixes, supervises student actions without environment calls, and samples earlier steps more often. Teacher trajectories can be reused from RL training.

et al., 2023; Nakano et al., 2021; Jin et al., 2025). Fully online multi-turn OPD reruns the student through the environment and queries the teacher at each visited history on every update. We instead study *prefix replay* (Figure 2): at a selected turn, the student generates an action from a teacher-recorded history and receives per-token teacher supervision along that action. Earlier actions and observations come from the recording, so no student action is executed in the environment. The supervised step is on-policy, while its roll-in is teacher-forced. Trajectories collected during teacher RL can be reused, and pools from separate environments can train one student without running those environments together (Appendix Figure 6). Figure 1 shows the accuracy and efficiency trade-off.

This reuse creates a *prefix trap*. Errors compound across turns, while student-generated histories improve relevance to the student but can make the teacher's target unreliable. Our bound separates the resulting student-occupancy and teacher-reliability errors (Section 2). It motivates choosing reliable prefix distributions rather than assuming fully student-on-policy roll-ins are always best. ReOPD implements this idea with step-decaying sampling that favors earlier, lower-shift teacher prefixes. When the teacher remains reliable on student histories, ordinary OPD is strong; as the teacher–student gap widens, teacher-supported prefixes can help.

Contributions.

- We introduce ReOPD, which reuses offline teacher trajectories for multi-turn distillation, and identify the *prefix trap*: compounding errors and a two-sided distribution shift.
- We bound the gap to an ideal interactive objective by student occupancy mismatch and teacher reliability, explaining why fully student-on-policy prefixes need not be optimal.
- We formulate prefix selection as a geometric bridge between student and teacher occupancies, motivating a reliability-aware step-decay schedule.
- Across mathematical reasoning with Python and search, multiple model scales, and joint training, ReOPD improves over SFT where reported. It outperforms online OPD on math when the teacher–student gap is wide and essentially matches OPD on search/QA, while having $> \times 4$ speedup without any interaction with the environment.

2 Problem Formulation and Analysis

We formulate multi-turn on-policy distillation in an interactive environment. At each OPD update, let $\pi_{\theta_{\text{old}}}$ denote the current student policy used for data collection, and let π_θ denote the student policy being optimized. Each input $x \sim \mathcal{D}$ induces T agent–environment decision steps.

At interaction step t, the agent observes an interaction history $H_t = (O_1, A_1, O_2, A_2, \ldots, A_{t-1}, O_t)$, where O_t is the current observation and A_s is the action taken at step s. Given H_t, a policy π chooses an action $A_t \sim \pi(\cdot \mid x, H_t)$, and the environment returns the next observation $O_{t+1} \sim \mathcal{E}(\cdot \mid x, H_t, A_t)$. Thus a step is an interaction step consisting of a policy action followed by an environment response, rather than merely a token-generation step.

For any policy π, the policy-environment system induces a step-t history occupancy distribution $d^t_{\pi,\mathcal{E}}(h_t \mid x) = \Pr_{\pi,\mathcal{E}}(H_t = h_t \mid x)$, which factorizes recursively: if $h_{t+1} = (h_t, a_t, o_{t+1})$, then

$$d^{t+1}_{\pi,\mathcal{E}}(h_{t+1} \mid x) = d^t_{\pi,\mathcal{E}}(h_t \mid x)\, \pi(a_t \mid x, h_t)\, \mathcal{E}(o_{t+1} \mid x, h_t, a_t).$$

In particular, the current student $\pi_{\theta_{\text{old}}}$ induces the student interaction occupancy $d^t_{\theta_{\text{old}}} \equiv d^t_{\pi_{\theta_{\text{old}}},\mathcal{E}}$, while the teacher π_T induces the teacher interaction occupancy $d^t_T \equiv d^t_{\pi_T,\mathcal{E}}$. Once a history h_t is collected, the teacher is queried for a target action distribution $\pi_T(\cdot \mid x, h_t)$. Therefore, in interactive OPD the design choice is not only which target distribution to distill from, but also which interaction histories to query the teacher on.

Ideal interactive improvement objective. Let $q^\star_t(\cdot \mid x, h_t)$ denote the ideal improvement target at interaction step t and history (x, h_t). The ideal objective evaluates the updated student π_θ on histories that the current student $\pi_{\theta_{\text{old}}}$ will actually encounter under the environment interaction:

$$\mathcal{R}^\star(\theta; \theta_{\text{old}}) = \mathbb{E}_{x\sim\mathcal{D}}\left[\sum_{t=1}^{T} \alpha_t\, \mathbb{E}_{H_t \sim d^t_{\theta_{\text{old}}}(\cdot|x)}\left[\ell\left(\pi_\theta(\cdot \mid x, H_t), q^\star_t(\cdot \mid x, H_t)\right)\right]\right], \tag{1}$$

where $\alpha_t \geq 0$ controls the relative importance of different interaction steps; unless a schedule is specified we take $\alpha_t \equiv 1$ (uniform across steps), so that ReOPD's reliability signal – including its step-decay – is carried entirely by the weight w_t (Section 3), not by α_t. This targets histories generated by the student's own interactions.

In practice, $q^\star_t$ is unavailable. OPD therefore replaces it with the teacher distribution $\pi_T(\cdot \mid x, h_t)$. However, the teacher is now queried on histories generated by a particular roll-in process. If those histories are far from the teacher's reliable interaction support, the teacher's action distribution may not be a trustworthy improvement signal.

From online interaction to offline reuse. The occupancies $d^t_{\pi,\mathcal{E}}$ and the ideal objective $\mathcal{R}^\star$ are defined through environment interaction, but they serve only as *conceptual targets*. Realizing them online would require, at every update, rolling the current student through the environment and querying the teacher afresh at each visited history – precisely the per-step environment and teacher cost we wish to avoid. We instead train off-environment from a fixed pool of pre-collected teacher trajectories: at each recorded history h_t the student proposes an action and the teacher conditional $\pi_T(\cdot \mid x, h_t)$ supplies the target, while no environment query is needed because the prefix and observations are replayed from the trace. Consequently the roll-in is the teacher pool, so the collected histories follow $\mathcal{P}_t \approx d^t_T$ rather than the student occupancy $d^t_{\theta_{\text{old}}}$. The student occupancy that the ideal objective targets is therefore *never sampled*; obtaining it is exactly what online interaction would buy. This is the crux of the offline setting: we must approximate an objective defined over $d^t_{\theta_{\text{old}}}$ using only samples from d^t_T, which is what the reweighting below accomplishes.

Collected and effective history distributions. Write $\mathcal{P}_t(\cdot \mid x)$ for the roll-in distribution histories are collected from; in the offline setting it is the fixed teacher pool, so $\mathcal{P}_t \approx d^t_T$. A general weighted OPD objective is

$$\mathcal{L}_{\mathcal{P},w}(\theta; \theta_{\text{old}}) = \mathbb{E}_{x\sim\mathcal{D}}\left[\sum_{t=1}^{T} \alpha_t\, \mathbb{E}_{H_t \sim \mathcal{P}_t(\cdot|x)}\left[w_t(x, H_t)\, \ell\left(\pi_\theta(\cdot \mid x, H_t), \pi_T(\cdot \mid x, H_t)\right)\right]\right], \tag{2}$$

where the weights are normalized per (x, t), i.e. $\mathbb{E}_{H_t\sim\mathcal{P}_t(\cdot|x)}[w_t(x, H_t)] = 1$, so that w_t reshapes $\mathcal{P}_t$ into the per-step distribution ρ_t; the position-only schedule of Section 3, constant across histories at each t, is instead normalized across positions, as detailed there. Then w_t induces an effective interaction-prefix distribution $\rho_t(h_t \mid x) = w_t(x, h_t)\, \mathcal{P}_t(h_t \mid x)$, and the objective becomes

$$\mathcal{L}_\rho(\theta; \theta_{\text{old}}) = \mathbb{E}_{x\sim\mathcal{D}}\left[\sum_{t=1}^{T} \alpha_t\, \mathbb{E}_{H_t \sim \rho_t(\cdot|x)}\left[\ell\left(\pi_\theta(\cdot \mid x, H_t), \pi_T(\cdot \mid x, H_t)\right)\right]\right]. \tag{3}$$

Thus the central design problem is not the raw weight w_t, but the effective history distribution ρ_t.

Two sources of mismatch. The practical objective differs from the ideal interactive improvement objective for two reasons. First, the effective history distribution ρ_t may differ from the student-environment occupancy $d^t_{\theta_{\text{old}}}$. Second, even when histories are relevant to the student, the teacher may be unreliable on histories far from its own interaction support.

Define the step-wise teacher reliability error

$$\epsilon^{\theta}_{T,t}(x, h_t) = \left|\ell\left(\pi_\theta(\cdot \mid x, h_t), q^\star_t(\cdot \mid x, h_t)\right) - \ell\left(\pi_\theta(\cdot \mid x, h_t), \pi_T(\cdot \mid x, h_t)\right)\right|.$$

Assumption 1 (Bounded improvement loss)**.** *The loss against the ideal target is uniformly bounded: there exists a constant $B < \infty$ such that $\left|\ell\left(\pi_\theta(\cdot \mid x, h_t), q^\star_t(\cdot \mid x, h_t)\right)\right| \le B$ for all (x, h_t, t).*

Assumption 1 is mild and covers the losses we use. It holds for any bounded divergence – e.g. total variation ($B = 1$) or Jensen–Shannon ($B = \log 2$). For the per-token KL used in Section 3, it holds whenever the target conditional is bounded below on its support, $q^\star_t(a \mid x, h_t) \ge p_{\min} > 0$, since then $D_{\text{KL}}(\pi_\theta \,\|\, q^\star_t) \le \log(1/p_{\min}) =: B$; a softmax with bounded logits (equivalently a temperature or label-smoothing floor) guarantees such a $p_{\min}$. Only the ideal-target loss needs to be bounded: the teacher-target loss enters the bound through the difference $\epsilon^{\theta}_{T,t}$.

Proposition 1 (Two-sided decomposition)**.** *Recall the ideal objective $\mathcal{R}^\star(\theta; \theta_{\text{old}})$ (1), which scores π_θ against the ideal target $q^\star_t$ on the student occupancy $d^t_{\theta_{\text{old}}}$, and the replayed objective $\mathcal{L}_\rho(\theta; \theta_{\text{old}})$ (3), which scores π_θ against the teacher π_T on the effective prefix distribution $\rho_t = w_t \mathcal{P}_t$. Under Assumption 1 (with bound B),*

$$\begin{aligned}\left|\mathcal{R}^\star(\theta; \theta_{\text{old}}) - \mathcal{L}_\rho(\theta; \theta_{\text{old}})\right| \le \mathbb{E}_{x\sim\mathcal{D}} \sum_{t=1}^{T} \alpha_t \\ \left[2B\,\text{TV}\big(d^t_{\theta_{\text{old}}}(\cdot \mid x), \rho_t(\cdot \mid x)\big) + \mathbb{E}_{H_t\sim\rho_t(\cdot|x)}\left[\epsilon^{\theta}_{T,t}(x, H_t)\right]\right].\end{aligned} \tag{4}$$

Here TV *is the total-variation distance, $\alpha_t \ge 0$ the step coefficients of (1), and $\epsilon^{\theta}_{T,t}(x, h_t) = \left|\ell(\pi_\theta, q^\star_t) - \ell(\pi_\theta, \pi_T)\right|$ the step-wise teacher-reliability error at (x, h_t).*

The proof is in Appendix B. The first term is the student interaction-occupancy mismatch: it is small when the effective training histories match the histories that the current student will encounter through environment interaction. The second term is the teacher reliability mismatch: it is small when the teacher's conditional is a reliable improvement target on the queried histories.

Fully student-on-policy OPD ($\rho_t = d^t_{\theta_{\text{old}}}$ for all t) zeroes the occupancy term but not the reliability one: student-generated actions can steer the environment into histories unlikely under the teacher's own interaction process, especially at later steps where its conditional is a poorly calibrated target. Teacher-forced histories ($\rho_t = d^t_T$) do the reverse – reliable near the teacher's own occupancy, but not where the student actually goes at test time. Neither extreme is uniformly optimal, so the effective distribution should balance student relevance against teacher reliability.

Reliability-aware interaction-prefix distribution. The preceding decomposition motivates choosing, for each interaction step, an effective history distribution

$$\rho^\star_t \in \arg\min_{\rho_t} \left\{\lambda_{\text{stu},t}\, D\left(\rho_t(\cdot \mid x), d^t_{\theta_{\text{old}}}(\cdot \mid x)\right) + \lambda_{\text{tea},t}\, \mathcal{E}_T(\rho_t; x)\right\},$$

where D measures mismatch to the current student's occupancy and $\mathcal{E}_T(\rho_t; x)$ measures expected teacher unreliability under ρ_t. Since the true reliability error is generally unobserved, we use teacher support as a practical surrogate, giving the bridge objective

$$\rho^\star_t \in \arg\min_{\rho_t} \left\{\lambda_{\text{stu},t}\, D_{\text{KL}}\left(\rho_t(\cdot \mid x) \,\|\, d^t_{\theta_{\text{old}}}(\cdot \mid x)\right) + \lambda_{\text{tea},t}\, D_{\text{KL}}\left(\rho_t(\cdot \mid x) \,\|\, d^t_T(\cdot \mid x)\right)\right\}. \tag{5}$$

When the supports overlap, the solution is the geometric bridge

$$\rho^\star_t(h_t \mid x) \propto \left[d^t_{\theta_{\text{old}}}(h_t \mid x)\right]^{\gamma_t} \left[d^t_T(h_t \mid x)\right]^{1-\gamma_t}, \qquad \gamma_t = \frac{\lambda_{\text{stu},t}}{\lambda_{\text{stu},t} + \lambda_{\text{tea},t}}. \tag{6}$$

When $\gamma_t = 1$ the bridge becomes fully student-on-policy; when $\gamma_t = 0$ it becomes teacher-supported roll-in. Intermediate values select histories that are both likely under the current student's environment interaction and not too far from the teacher's reliable interaction support.

Two regimes. The bound (4) sums two competing terms: the occupancy term is minimized by $\gamma_t \to 1$ (fully student-on-policy) and the reliability term by $\gamma_t \to 0$ (teacher-supported roll-in), so which one dominates sets the right γ_t. Crucially the balance is *per step*: the teacher-reliability term grows with depth, as roll-ins drift into histories the teacher rarely visits, so the ideal γ_t *decreases along the trajectory* – student-relevant ($\gamma_t \to 1$) at early steps, teacher-supported ($\gamma_t \to 0$) only at the deep steps where the teacher signal has degraded. This yields two regimes. When the teacher stays reliable on student-induced histories – a capable teacher close to the student, short horizons, or early steps – the occupancy term dominates, $\gamma_t \to 1$ is best, and ordinary student-on-policy OPD is already strong. When the teacher's signal degrades off its own support – a large teacher–student gap, long horizons, or later steps – lowering γ_t *at those steps* reduces the gap even though the resulting histories are *less* student-on-policy, which is counterintuitive if one reads OPD as merely "make it on-policy". Section 3 approximates this depth-dependent retreat without tuning γ_t directly – shifting supervision mass toward earlier steps rather than reshaping the roll-in within each step – applying on the fixed teacher-pool roll-in a step-decay whose steepness grows with the teacher–student gap.

From distribution to weight. When histories come from $\mathcal{P}_t$, any effective distribution $\rho_t^\star$ is realized by the normalized importance weight $w_t^\star = \rho_t^\star / \mathcal{P}_t$ (with $\mathbb{E}_{H_t \sim \mathcal{P}_t}[w_t^\star] = 1$), which recovers the weighted objective (2) and can be applied equivalently by reweighting or by sampling prefixes in proportion to it. The weight is thus an implementation device, not the conceptual core: the principle is to choose, at each step, an effective history distribution that balances student relevance and teacher reliability. Because exact density ratios are high-variance, the concrete step-decay schedule we use in practice is developed in Section 3. After optimizing the weighted objective the student is refreshed, $\theta_{\text{new}} \leftarrow \arg\min_\theta \mathcal{L}_\rho(\theta; \theta_{\text{old}})$, and the next OPD iteration sets $\theta_{\text{old}} \leftarrow \theta_{\text{new}}$.

Interpretation. On-policy distillation is not automatically optimal merely because histories are sampled from the student: the history distribution sets both where the student is trained and where the teacher is asked to supervise, so when the teacher is unreliable on student-induced histories, a reliability-aware distribution close to both occupancies is the safer target.

3 Replayed-Prefix On-Policy Distillation

We now turn that principle into a concrete algorithm, *Replayed-Prefix On-Policy Distillation* (ReOPD, Algorithm 1), realized off-environment from a fixed pool of teacher trajectories. The analysis reduces multi-turn OPD to a single design choice: at each step, pick an effective prefix distribution ρ_t that balances student relevance against teacher reliability. ReOPD has two components:

1. **Teacher-forced prefix, student action, teacher supervision.** For a supervised step t, the prefix $h_t = (O_1, A_1, \ldots, O_t)$ is taken *entirely* from a pre-collected teacher trajectory – every earlier action and every observation is the teacher's recorded one. At this prefix the *student* produces its own action, and the teacher's recorded conditional $\pi_T(\cdot \mid x, h_t)$ supplies the distillation target. Sweeping t over the trajectory covers all positions without querying the environment.
2. **Reliability-aware step schedule.** The student is on-policy only at the evaluated step, on top of a teacher-forced prefix. As the position moves deeper into the trajectory, the replayed teacher prefix becomes a higher-shift surrogate for the histories the student would have reached. A per-step schedule therefore emphasizes early, low-shift positions and de-emphasizes late, high-shift positions while keeping the teacher conditional as the distillation target.

The point is that on-policy supervision and off-environment training coexist: the student's step-t action makes the target relevant to the student, while the fully teacher-forced prefix keeps the roll-in within the teacher's occupancy ($\mathcal{P}_t \approx d_T^t$) and removes any environment interaction. The step schedule controls how these teacher-supported histories stand in for the student occupancy $d_{\theta_{\text{old}}}^t$.

Inputs and notation. We are given a teacher π_T and a pool $\mathcal{D}_T = \{(x, h)\}$ of trajectories generated by the teacher interacting with the environment, where each h records the full interaction history including the environment observations. In practice this pool requires no dedicated collec-

tion: when the teacher is trained with on-policy RL, its training rollouts are exactly such trajectories, so $\mathcal{D}_T$ is a free by-product reused for distillation.

Off-environment prefix construction. For each supervised step t, the prefix $h_t = (O_1, A_1, \ldots, O_t)$ is replayed *verbatim* from the teacher trace – all earlier actions $A_{<t}$ and all observations $O_{\leq t}$ are the teacher's. The student only acts at step t, where it generates its own action $A_t = (a_t^1, \ldots, a_t^{n_t}) \sim \pi_{\theta_{\text{old}}}(\cdot \mid x, h_t)$ autoregressively and is supervised, *per generated token*, by the teacher's conditional along that action, $\pi_T(\cdot \mid x, h_t, a_t^{<j})$. Because the supervised token contexts $a_t^{<j}$ are the student's own samples, the evaluated step is genuinely on-policy even though the prefix is teacher-forced, and it enters the loss below through these on-policy contexts. No action is ever executed against the environment and no observation is ever generated, so no environment is queried. The roll-in is thus the teacher occupancy, $\mathcal{P}_t \approx d_T^t$; the two-sided shift arises because these replayed prefixes are only a surrogate for the histories the current student would reach; the schedule below emphasizes earlier positions where the two occupancies are closer.

Reliability-aware step schedule. The exact bridge (6) weights each recorded history by a student-to-teacher likelihood ratio along its prefix. This ratio is expensive and high-variance, and its magnitude typically decreases as the teacher-recorded prefix grows longer. ReOPD uses the step index as a stable proxy: $\omega(t;\kappa) = \kappa^t$, with $\kappa \in (0, 1]$, so early, lower-shift positions receive more training mass. This schedule changes the relative *frequency of supervised positions*, not the conditional distribution of histories within a position. The derivation, exact history-level alternative, and measured ratio decay appear in Appendix C.

Weighted distillation objective. At each supervised step the student samples its own action $A_t = (a_t^1, \ldots, a_t^{n_t}) \sim \pi_{\theta_{\text{old}}}(\cdot \mid x, H_t)$ and is distilled against the teacher *along that action*: the per-step loss is the token-level KL accumulated over the student-sampled action tokens,

$$\ell\big(\pi_\theta, \pi_T; x, H_t, A_t\big) = \sum_{j=1}^{n_t} D_{\text{KL}}\Big(\pi_\theta(\cdot \mid x, H_t, a_t^{<j}) \,\Big\|\, \pi_T(\cdot \mid x, H_t, a_t^{<j})\Big),$$

whose conditioning contexts $a_t^{<j}$ are on-policy, so the loss depends on the student's action. The student then minimizes the reweighted step-wise objective

$$\mathcal{L}(\theta) = \mathbb{E}_{x\sim\mathcal{D}}\left[\sum_{t=1}^{T} \alpha_t\, \mathbb{E}_{H_t\sim\mathcal{P}_t(\cdot|x)}\, \mathbb{E}_{A_t\sim\pi_{\theta_{\text{old}}}(\cdot|x,H_t)}\Big[w_t\, \ell\big(\pi_\theta, \pi_T; x, H_t, A_t\big)\Big]\right].$$

Here $\alpha_t \equiv 1$ is the uniform base of Section 2; the schedule $w_t = \omega(t;\kappa) = \kappa^t$ reallocates mass across positions. We implement it by sampling positions with probability proportional to w_t and applying the unweighted loss, rather than enumerating every position and weighting its loss. Both estimate the same horizon-reweighted objective (Appendix C). After optimization the student is refreshed, $\theta_{\text{old}} \leftarrow \theta$, and the next iteration re-evaluates the student's steps against the same teacher prefixes, so supervision tracks the moving student while staying anchored to the teacher pool.

Setting the decay. The only free knob is the steepness κ. The map derived in Appendix C makes it measurable: $\kappa \approx \exp(-\gamma_t \bar{c})$ is read off from the per-step teacher–student gap $\bar{c}$, an average $D_{\text{KL}}(\pi_T \| \pi_{\theta_{\text{old}}})$ estimated directly on the pool. The weight always decreases with depth, so early, low-shift steps receive the most mass and κ only sets how fast it falls off: $\kappa = 1$ is uniform (OPD-like) and smaller κ concentrates supervision on the early steps. The two regimes read off through $\bar{c}$ – a small gap gives $\kappa \approx 1$ and gentle decay, a large gap (e.g. the wide teacher–student math setting) gives small κ and steep decay – and steepness is set empirically in Section 4, entering only through the sampling probability $p_t \propto w_t$ (equivalently, a weight w_t).

Why this is more than "make it on-policy". Standard OPD pushes prefixes toward the student by re-rolling through a live environment; ReOPD instead replays a teacher-forced prefix and corrects with a reliability-aware step-decay schedule, achieving student relevance off-environment. The principle is reliability-aware prefix selection, implemented with step-decaying sampling.

Algorithm 1 Replayed-Prefix On-Policy Distillation (ReOPD)

Require: teacher pool $\mathcal{D}_T$, teacher π_T, student π_θ, step-decay schedule $\omega(t;\kappa) = \kappa^t$

initialize $\theta_{\text{old}} \leftarrow \theta$
for each OPD iteration **do**
  assemble candidate positions $\{(x, h_t)\}$ from pooled trajectories ▷ prefix h_t: teacher actions $A_{<t}$, teacher observations $O_{\leq t}$
  draw positions with $p_t \propto \omega(t;\kappa)$ ▷ early, low-shift steps sampled more often
  for each sampled position (x, h_t) **do**
    student generates its action $A_t = (a_t^1, \ldots, a_t^{n_t}) \sim \pi_{\theta_{\text{old}}}(\cdot \mid x, h_t)$ ▷ token contexts on-policy; no environment call
    accumulate $\sum_{j=1}^{n_t} D_{\text{KL}}\big(\pi_\theta(\cdot \mid x, h_t, a_t^{<j}) \,\|\, \pi_T(\cdot \mid x, h_t, a_t^{<j})\big)$ ▷ per-token KL along the student's action A_t
  end for
  update θ on the accumulated loss; $\theta_{\text{old}} \leftarrow \theta$
end for
return π_θ

Table 1: **Distillation on math.** Accuracy on six Python-tool benchmarks; Avg. is their mean. GRPO is the teacher's RL result, and the better OPD/ReOPD score in each column is bold.

Method	AIME24	AIME25	AMC23	Minerva	Olympiad	MATH500	*Avg.*
Teacher model: Qwen3-4B-Instruct-2507							
GRPO	42.1	32.1	79.7	37.9	56.1	85.1	55.5
Student model: Qwen3-4B-Instruct-2507							
Base	6.3	6.7	36.6	31.0	29.5	58.2	28.0
Cold Start	22.1	20.8	66.6	32.4	49.0	79.1	45.0
SFT	21.7	21.3	63.1	39.8	49.9	80.9	46.1
OPD	35.4	29.2	77.8	**43.3**	58.2	86.9	55.1
ReOPD	**40.8**	**31.3**	**80.0**	42.9	**59.4**	**88.9**	**57.2**
Teacher model: Qwen3-8B							
GRPO	28.3	23.3	73.1	38.5	52.0	84.3	49.9
Student model: Qwen3-4B-Instruct-2507							
SFT	22.1	22.9	63.8	36.0	44.7	80.8	45.0
OPD	28.3	26.3	70.3	39.8	55.3	**85.8**	51.0
ReOPD	**36.7**	**29.2**	**74.4**	**41.1**	**56.4**	84.7	**53.7**
Teacher model: Qwen3-30B-A3B-Instruct-2507							
GRPO	47.9	35.0	91.3	46.4	63.7	92.1	62.7
Student model: Qwen3-4B-Instruct-2507							
OPD	28.3	25.8	**75.3**	**39.6**	53.6	83.8	51.1
ReOPD	**32.5**	**26.7**	74.4	39.4	**55.8**	**86.5**	**52.5**
Student model: Qwen3-8B							
Base	21.3	11.3	43.1	37.2	34.4	64.8	35.3
Cold Start	23.9	19.6	67.8	38.6	49.0	82.0	46.8
OPD	40.8	29.2	77.8	**45.1**	**56.9**	**89.0**	56.5
ReOPD	**41.7**	**30.4**	**80.6**	44.1	55.7	88.4	**56.8**

4 EXPERIMENTS

We evaluate a Qwen3-4B student against Qwen3-4B, Qwen3-8B, and Qwen3-30B-A3B teachers in math with Python and search environments. Teachers are trained with GRPO; students start from supervised fine-tuning and are distilled using either fully online OPD or ReOPD with the same teacher targets and update budget. ReOPD samples steps with $\kappa = 0.6$ throughout. Evaluation benchmarks, baselines, training hyperparameters, and dataset construction are detailed in Appendix D.

Table 2: **Distillation on search.** Per-benchmark accuracy on general and multi-hop QA; † marks in-domain benchmarks and * out-of-domain. GRPO is the teacher's RL result.

Method	General QA			Multi-Hop QA				*Avg.*
	NQ†	TriviaQA*	PopQA*	HotpotQA†	2wiki*	Musique*	Bamboogle*	
Teacher model: Qwen3-4B-Instruct-2507								
GRPO	36.1	59.2	40.2	37.2	31.8	13.8	39.2	40.4
Student model: Qwen3-4B-Instruct-2507								
Base	28.4	55.5	30.6	30.8	**33.6**	8.4	35.8	35.6
Cold Start	26.2	48.6	33.6	24.7	26.6	5.8	24.8	32.1
OPD	36.3	**59.7**	40.3	**37.3**	31.7	**14.5**	**44.0**	**40.6**
ReOPD	**36.4**	59.6	**40.5**	**37.3**	31.5	14.0	37.6	40.5
Teacher model: Qwen3-8B								
GRPO	36.6	59.3	40.2	36.7	34.7	13.8	39.2	41.0
Student model: Qwen3-4B-Instruct-2507								
OPD	**35.5**	57.3	39.0	**35.0**	**31.2**	**12.8**	**37.6**	**39.1**
ReOPD	34.9	**57.5**	**39.1**	34.6	31.0	11.9	**37.6**	39.0

Table 3: **Joint math and search training.** One Qwen3-4B student distilled from separate domain-specific Qwen3-4B teachers. Search benchmarks marked † are in-domain and * are out-of-domain; GRPO shows each teacher's result.

Benchmark	Teacher (Qwen3-4B)	Student (Qwen3-4B)		
	GRPO	Cold Start	OPD	ReOPD
Mathematical reasoning				
AIME24	42.1	18.8	34.2	**36.3**
AIME25	32.1	23.8	29.6	**30.8**
AMC23	79.7	67.8	**79.7**	75.5
Minerva	37.9	40.5	**44.5**	**44.5**
Olympiad	56.1	48.9	57.1	**57.4**
MATH500	85.1	82.4	**87.3**	**87.3**
Avg.	55.5	47.0	**55.4**	55.3
Search				
NQ†	36.1	25.7	37.3	**37.7**
TriviaQA*	59.2	51.3	**61.0**	60.7
PopQA*	40.2	33.2	40.5	**40.6**
HotpotQA†	37.2	25.1	**38.0**	37.6
2wiki*	31.8	27.1	31.6	**32.1**
Musique*	13.8	6.3	**14.5**	13.8
Bamboogle*	39.2	29.6	**44.0**	37.6
Avg.	40.4	32.7	**41.0**	**41.0**

4.1 Main Results

Single environment. Table 1 shows that ReOPD improves the student average in all four teacher–student settings: $55.1 \rightarrow 57.2$ (4B/4B), $51.0 \rightarrow 53.7$ (8B/4B), $51.1 \rightarrow 52.5$ (30B-A3B/4B), and $56.5 \rightarrow 56.8$ (30B-A3B/8B). The larger gains for 4B students are consistent with teacher-supported prefixes helping when student roll-ins drift.

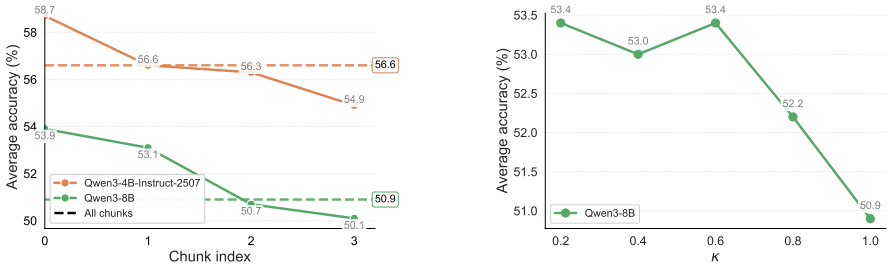
On search, ReOPD essentially matches OPD: 40.5 vs. 40.6 with the 4B teacher and 39.0 vs. 39.1 with the 8B teacher (Table 2). Teacher-anchored replay thus preserves accuracy when the teacher remains reliable on student histories.

Multiple environments. Table 3 shows that one Qwen3-4B student distilled jointly from domain-specific math and search teachers matches OPD (math 55.3 vs. 55.4; search 41.0 each) without live environments during student training.

Efficiency. ReOPD uses zero student-training tool calls and is at least $4\times$ faster per rollout while preserving OPD-level accuracy (Figure 1). Further analyses are in Appendix E.

Table 4: **RL rollout reuse.** Final Qwen3-8B target; Qwen3-4B student.

Prefix pool	AIME24	AIME25	AMC23	Minerva	Olympiad	MATH500	*Avg.*
Final-teacher (stationary)	37.9	30.0	72.2	39.2	56.2	85.0	53.4
RL rollouts (free, mixed)	36.7	29.2	74.4	41.1	56.4	84.7	53.7



(a) Sampling by chunk index.

(b) Sampling with decay κ.

Figure 3: **Early-step sampling improves distillation.** **(a)** Earlier chunks help; **(b)** moderate κ^t decay beats uniform sampling. Student: Qwen3-4B-Instruct-2507.

Ablations. Earlier steps outperform uniform sampling under both chunk-based and decay-weighted selection (Figure 3). As step index grows, teacher prefixes become less likely under the student (Appendix Figure 5), making position a proxy for prefix shift. With the teacher target fixed, its own prefixes outperform those from stronger generators (Figure 4). These results favor teacher-supported prefixes, though step index cannot identify each reliable history.

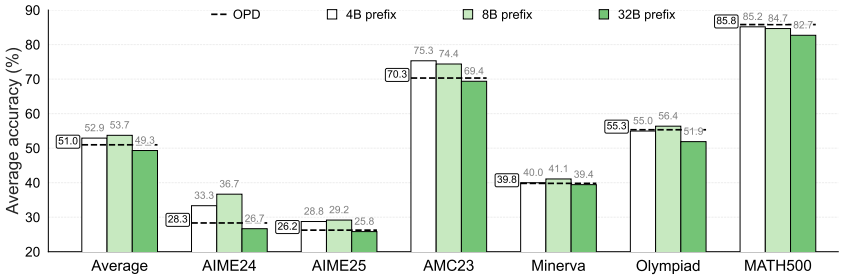


Figure 4: **Teacher prefixes beat stronger generators.** Qwen3-8B teacher, Qwen3-4B student.

Reusing teacher rollouts. Mixed RL checkpoints match a fresh final-teacher pool (53.7 vs. 53.4; Table 4). Without RL traces, one-time teacher collection still permits off-environment student training and exceeds OPD speed by 2× per rollout, including collection (Appendix Figure 8).

5 Related Work

Sequence distillation trains on teacher traces (Kim & Rush, 2016), whereas OPD queries teachers on student prefixes (Agarwal et al., 2024). DAgger counters learner-state shift (Ross et al., 2011), but multi-turn roll-ins need live environments and can leave teacher support. ReOPD replays teacher prefixes for dense supervision rather than using rewards or filtered successes (more in Appendix A).

6 Conclusion

ReOPD distills student actions on replayed teacher prefixes without environment calls. The *prefix trap* combines compounding errors with unreliable teacher targets on shifted histories. Our bound (4) separates occupancy mismatch from teacher reliability and motivates early sampling. ReOPD matches OPD with reliable targets and improves as the teacher–student gap grows. Step index only approximates reliability; history-dependent selection may help. Teacher-trajectory coverage limits the method (Appendix F).

AI USE STATEMENT

Generative AI tools assisted with organizing and editing the manuscript, drafting parts of its theoretical exposition and interpretation of the reported experimental results, and preparing the LaTeX layout. The authors are responsible for the final claims, proofs, references, and numerical results.

REFERENCES

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

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms. *arXiv preprint arXiv:2402.14740*, 2024.

Renat Aksitov, Sobhan Miryoosefi, Zonglin Li, Daliang Li, Sheila Babayan, Kavya Kopparapu, Zachary Fisher, Ruiqi Guo, Sushant Prakash, Pranesh Srinivasan, et al. Rest meets react: Self-improvement for multi-step reasoning llm agent. *arXiv preprint arXiv:2312.10003*, 2023.

Thomas Anthony, Zheng Tian, and David Barber. Thinking fast and slow with deep learning and tree search. *Advances in neural information processing systems*, 30, 2017.

Art of Problem Solving Wiki. 2024 AIME Problems and Solutions. https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions, 2024.

Art of Problem Solving Wiki. 2025 AIME Problems and Solutions. https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions, 2025.

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. A general theoretical paradigm to understand learning from human preferences. *arXiv preprint arXiv:2310.12036*, 2023.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, 2015.

Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Baian Chen, Chang Shu, Ehsan Shareghi, Nigel Collier, Karthik Narasimhan, and Shunyu Yao. Fireact: Toward language agent fine-tuning. *arXiv preprint arXiv:2310.05915*, 2023.

Zehui Chen, Kuikun Liu, Qiuchen Wang, Wenwei Zhang, Jiangning Liu, Dahua Lin, Kai Chen, and Feng Zhao. Agent-flan: Designing data and methods of effective agent tuning for large language models. *arXiv preprint arXiv:2403.12881*, 2024.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai

Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, KaShun SHUM, and Tong Zhang. RAFT: Reward ranked finetuning for generative foundation model alignment. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=m7p5O7zblY.

Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen Sahoo, Caiming Xiong, and Tong Zhang. Rlhf workflow: From reward modeling to online rlhf. *arXiv preprint arXiv:2405.07863*, 2024.

Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. Retool: Reinforcement learning for strategic tool use in llms. *arXiv preprint arXiv:2504.11536*, 2025.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yujiu Yang, Minlie Huang, Nan Duan, Weizhu Chen, et al. Tora: A tool-integrated reasoning agent for mathematical problem solving. *arXiv preprint arXiv:2309.17452*, 2023.

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

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, et al. Reinforced self-training (rest) for language modeling. *arXiv preprint arXiv:2308.08998*, 2023.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng 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. *arXiv preprint arXiv:2402.14008*, 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.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*, 2020.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics (ACL Findings)*, 2023.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2017.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pp. 6769–6781, 2020.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1317–1327, 2016.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. Distillm: Towards streamlined distillation for large language models. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 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. *arXiv preprint arXiv:2503.07067*, 2025.

Wouter Kool, Herke van Hoof, and Max Welling. Buy 4 reinforce samples, get a baseline for free! In *ICLR Workshop on Deep Reinforcement Learning Meets Structured Prediction*, 2019. URL https://wouterkool.github.io/publication/buy-4-samples-free-baseline/.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019.

Alex Lamb, Anirudh Goyal, Ying Zhang, Saizheng Zhang, Aaron Courville, and Yoshua Bengio. Professor forcing: A new algorithm for training recurrent networks. In *Advances in Neural Information Processing Systems*, 2016.

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.

Ziniu Li, Tian Xu, Yushun Zhang, Zhihang Lin, Yang Yu, Ruoyu Sun, and Zhi-Quan Luo. Remax: A simple, effective, and efficient reinforcement learning method for aligning large language models. *arXiv preprint arXiv:2310.10505*, 2023.

Baohao Liao, Hanze Dong, Xinxing Xu, Christof Monz, and Jiang Bian. Self-hinting language models enhance reinforcement learning. *arXiv preprint arXiv:2602.03143*, 2026.

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. *arXiv preprint arXiv:2305.20050*, 2023.

Tianqi Liu, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter J Liu, and Jialu Liu. Statistical rejection sampling improves preference optimization. *arXiv preprint arXiv:2309.06657*, 2023.

Kevin Lu and Thinking Machines Lab. On-policy distillation. https://thinkingmachines.ai/blog/on-policy-distillation/, 2025. Thinking Machines Lab: Connectionism.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023.

Arindam Mitra, Hamed Khanpour, Corby Rosset, and Ahmed Awadallah. Orca-math: Unlocking the potential of slms in grade school math. *arXiv preprint arXiv:2402.14830*, 2024.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744, 2022.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*, 2023.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. In *The Fourth International Conference on Learning Representations (ICLR)*, 2016.

Stéphane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 627–635, 2011.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

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, Mingchuan Zhang, YK Li, Y Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.

Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Peter J Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron Parisi, et al. Beyond human data: Scaling self-training for problem-solving with language models. *arXiv preprint arXiv:2312.06585*, 2023.

Zetian Sun, Dongfang Li, Xuhui Chen, Baotian Hu, and Min Zhang. Is on-policy data always the best choice for direct preference optimization-based lm alignment? *arXiv preprint arXiv:2508.10530*, 2025.

Fahim Tajwar, Anikait Singh, Archit Sharma, Rafael Rafailov, Jeff Schneider, Tengyang Xie, Stefano Ermon, Chelsea Finn, and Aviral Kumar. Preference fine-tuning of llms should leverage suboptimal, on-policy data. *arXiv preprint arXiv:2404.14367*, 2024.

Yunhao Tang, Zhaohan Daniel Guo, Zeyu Zheng, Daniele Calandriello, Rémi Munos, Mark Rowland, Pierre Harvey Richemond, Michal Valko, Bernardo Ávila Pires, and Bilal Piot. Generalized preference optimization: A unified approach to offline alignment. *arXiv preprint arXiv:2402.05749*, 2024.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics (TACL)*, 10, 2022.

Bing Wang, Shaotian Yan, Chen Shen, Kaiyuan Liu, Sinan Fan, Ximing Li, Rui Miao, Xiaosong Yuan, Zhanming Shen, and Jieping Ye. Backtracking when it strays: Mitigating dual exposure biases in llm reasoning distillation. *arXiv preprint arXiv:2605.19433*, 2026.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. *arXiv preprint arXiv:2312.08935*, 2023.

Xingyao Wang, Zihan Wang, Jiateng Liu, Yangyi Chen, Lifan Yuan, Hao Peng, and Heng Ji. Mint: Evaluating llms in multi-turn interaction with tools and language feedback. In *Proc. The Twelfth International Conference on Learning Representations (ICLR2024)*, 2024.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256, 1992.

Wei Xiong, Hanze Dong, Chenlu Ye, Ziqi Wang, Han Zhong, Heng Ji, Nan Jiang, and Tong Zhang. Iterative preference learning from human feedback: Bridging theory and practice for rlhf under kl-constraint. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 54715–54754, 2024. URL https://proceedings.mlr.press/v235/xiong24a.html.

Wei Xiong, Jiarui Yao, Yuhui Xu, Bo Pang, Lei Wang, Doyen Sahoo, Junnan Li, Nan Jiang, Tong Zhang, Caiming Xiong, and Hanze Dong. A minimalist approach to llm reasoning: from rejection sampling to reinforce. *arXiv preprint arXiv:2504.11343*, 2025.

Wenda Xu, Rujun Han, Zifeng Wang, Long T. Le, Dhruv Madeka, Lei Li, William Yang Wang, Rishabh Agarwal, Chen-Yu Lee, and Tomas Pfister. Speculative knowledge distillation: Bridging the teacher-student gap through interleaved sampling. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025a.

Yuhui Xu, Hanze Dong, Lei Wang, Doyen Sahoo, Junnan Li, and Caiming Xiong. Scalable chain of thoughts via elastic reasoning. *arXiv preprint arXiv:2505.05315*, 2025b.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Bill Yuchen Lin, and Radha Poovendran. Stronger models are not stronger teachers for instruction tuning. *arXiv preprint arXiv:2411.07133*, 2024.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

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.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2369–2380, 2018.

Jiarui Yao, Yifan Hao, Hanning Zhang, Hanze Dong, Wei Xiong, Nan Jiang, and Tong Zhang. Optimizing chain-of-thought reasoners via gradient variance minimization in rejection sampling and rl. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, pp. 163245–163284, 2025.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Tianzhu Ye, Li Dong, Zewen Chi, Xun Wu, Shaohan Huang, and Furu Wei. Black-box on-policy distillation of large language models. *arXiv preprint arXiv:2511.10643*, 2025.

Tianzhu Ye, Li Dong, Qingxiu Dong, Xun Wu, Shaohan Huang, and Furu Wei. Online experiential learning for language models. *arXiv preprint arXiv:2603.16856*, 2026a.

Tianzhu Ye, Li Dong, Xun Wu, Shaohan Huang, and Furu Wei. On-policy context distillation for language models. *arXiv preprint arXiv:2602.12275*, 2026b.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*, 2023.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Xiang Yue, Ge Zhang Xingwei Qu, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. Mammoth: Building math generalist models through hybrid instruction tuning. *arXiv preprint arXiv:2309.05653*, 2023.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Self-taught reasoner bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.

Hanning Zhang, Pengcheng Wang, Shizhe Diao, Yong Lin, Rui Pan, Hanze Dong, Dylan Zhang, Pavlo Molchanov, and Tong Zhang. Entropy-regularized process reward model, 2024. URL https://arxiv.org/abs/2412.11006.

Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J Liu. Slic-hf: Sequence likelihood calibration with human feedback. *arXiv preprint arXiv:2305.10425*, 2023.

Zirui Zhao, Hanze Dong, Amrita Saha, Caiming Xiong, and Doyen Sahoo. Automatic curriculum expert iteration for reliable llm reasoning. *arXiv preprint arXiv:2410.07627*, 2024.

Zilin Zhu, Chengxing Xie, Xin Lv, and slime Contributors. slime: An llm post-training framework for rl scaling. https://github.com/THUDM/slime, 2025. GitHub repository. Corresponding author: Xin Lv.

A Related Work

Knowledge distillation for language models. Knowledge distillation transfers a teacher's behavior into a smaller student (Hinton et al., 2015), and sequence-level distillation extends this idea to autoregressive generation by training on teacher trajectories (Kim & Rush, 2016). It underlies many strong small instructions and reasoning models trained from teacher-written solutions or rationales (Mitra et al., 2024; Yue et al., 2023; Yu et al., 2023; Hsieh et al., 2023). The standard recipe is efficient but off-policy: the student learns from the teacher prefixes, then must recover from its

own prefixes at inference. On-policy distillation addresses this by querying the teacher on student-sampled prefixes: MiniLLM uses reverse KL (Gu et al., 2024), GKD mixes student and teacher generations (Agarwal et al., 2024), DistiLLM refines the divergence and replay design (Ko et al., 2024; 2025), speculative distillation interleaves teacher and student tokens (Xu et al., 2025a), and recent practice frames OPD as dense supervision compared with sparse RL (Lu & Thinking Machines Lab, 2025). Recent extensions study black-box OPD and on-policy context distillation (Ye et al., 2025; 2026b). By analogy, a stronger teacher is not always a better teacher (Xu et al., 2024); there the concern is teacher and data selection for single-turn instruction tuning, whereas our reliability notion concerns *where* the teacher's per-step conditional is queried, so we take it as a suggestive parallel rather than direct support. We extend this line from single-turn generation to multi-turn training from a fixed teacher-trajectory pool, where the key object is not only the teacher target but the prefix distribution on which it is queried.

RL algorithms for LLM post-training. RL post-training optimizes scalar feedback rather than teacher conditionals. RLHF fits preference rewards, often Bradley–Terry models (Bradley & Terry, 1952), and optimizes them with PPO (Schulman et al., 2017), as in instruction-following and helpful-harmless assistants (Ouyang et al., 2022; Bai et al., 2022). Direct preference objectives simplify this pipeline by replacing online reward optimization with contrastive or implicit-reward losses such as SLIC-HF (Zhao et al., 2023), DPO (Rafailov et al., 2023), IPO (Azar et al., 2023), and GPO (Tang et al., 2024), with iterative and online variants studied under KL-regularized preference learning (Xiong et al., 2024; Dong et al., 2024). For reasoning, o1/DeepSeek-R1-style systems increasingly use verifiable or process rewards (Jaech et al., 2024; DeepSeek-AI et al., 2025; Wang et al., 2023; Zhang et al., 2024), giving rise to scalable methods such as GRPO (Shao et al., 2024) and DAPO (Yu et al., 2025). Critic-free variants revisit REINFORCE (Williams, 1992), including ReMax (Li et al., 2023), RLOO-style baselines (Ahmadian et al., 2024; Kool et al., 2019), and Reinforce-Rej (Xiong et al., 2025); RAFT and rejection sampling are the binary-reward limit, where successful trajectories are selected and imitated (Dong et al., 2023; Liu et al., 2023; Xiong et al., 2025). ReOPD is orthogonal to this optimizer family: it keeps dense teacher supervision and asks which replayed prefixes make that supervision reliable.

Reasoning and agentic LLMs. RL-style post-training also extends to agents that interact with tools and environments (Yao et al., 2022; Schick et al., 2023; Nakano et al., 2021), including search-augmented agents (Jin et al., 2025) and tool-integrated mathematical reasoning (Gou et al., 2023; Wang et al., 2024). Other work improves reasoning through self-hinting (Liao et al., 2026), budget-aware elastic reasoning (Xu et al., 2025b), online experiential learning (Ye et al., 2026a), or distills teacher-generated reasoning-and-acting traces into smaller agents (Chen et al., 2023; 2024). These works motivate our setting but optimize scalar rewards, advantages, filtered rollouts, or fixed teacher trajectories. We retain dense distillation targets without live environment interaction and instead select the replayed prefixes used for training.

Self-training and rejection sampling. Self-training methods repeatedly turn model samples into new training data. Expert-iteration methods imitate successful search or sampling solutions (Anthony et al., 2017), STaR bootstraps from self-generated rationales (Zelikman et al., 2022), and ReST-style methods alternate generation, scalar-feedback filtering, and retraining (Gulcehre et al., 2023; Singh et al., 2023). In LLM post-training, rejection sampling and RAFT instantiate the same principle by imitating reward-ranked trajectories (Dong et al., 2023; Liu et al., 2023; Xiong et al., 2025), with refinements for variance, curricula, and agentic self-improvement (Yao et al., 2025; Zhao et al., 2024; Aksitov et al., 2023); reflective methods revise trajectories from feedback (Shinn et al., 2023). ReOPD also reuses offline traces, but the trace is not merely accepted or rejected: it provides the histories on which the teacher is queried.

Distribution shift and compounding errors. The prefix trap inherits the classic sequential-learning problem: behavior cloning on expert states suffers covariate shift once the learner deviates, which DAgger addresses by training on the learner's induced state distribution (Ross et al., 2011). Autoregressive generation has the same exposure-bias structure; scheduled sampling, sequence-level training, and Professor Forcing reduce the mismatch between teacher-forced training and free-running inference (Bengio et al., 2015; Ranzato et al., 2016; Lamb et al., 2016). LLM alignment revisits this tension at scale: on-policy, even suboptimal, samples can help preference fine-

tuning (Tajwar et al., 2024), but on-policy data is not uniformly best and depends on the alignment stage (Sun et al., 2025). Recent reasoning distillation similarly identifies a dual exposure bias: teacher-forced traces mismatch student inference, while fully student-generated contexts can make teacher intervention unreliable (Wang et al., 2026). Our two-sided shift captures this trade-off between learner-relevant histories and teacher-supported supervision.

B Proof of the Two-Sided Bound

Proof of Proposition 1. Fix x and t, and abbreviate the ideal-target and teacher-target losses by $f_t(h) = \ell(\pi_\theta(\cdot \mid x, h), q_t^\star(\cdot \mid x, h))$ and $g_t(h) = \ell(\pi_\theta(\cdot \mid x, h), \pi_T(\cdot \mid x, h))$, so that $\mathcal{R}^\star = \mathbb{E}_x\big[\sum_t \alpha_t \, \mathbb{E}_{d^t_{\theta_{\text{old}}}}[f_t]\big]$ and $\mathcal{L}_\rho = \mathbb{E}_x\big[\sum_t \alpha_t \, \mathbb{E}_{\rho_t}[g_t]\big]$. Adding and subtracting $\mathbb{E}_{\rho_t}[f_t]$,

$$\mathbb{E}_{d^t_{\theta_{\text{old}}}}[f_t] - \mathbb{E}_{\rho_t}[g_t] = \underbrace{\big(\mathbb{E}_{d^t_{\theta_{\text{old}}}}[f_t] - \mathbb{E}_{\rho_t}[f_t]\big)}_{\text{occupancy shift}} + \underbrace{\big(\mathbb{E}_{\rho_t}[f_t] - \mathbb{E}_{\rho_t}[g_t]\big)}_{\text{teacher reliability}}.$$

For the occupancy-shift term, Assumption 1 gives $\|f_t\|_\infty \le B$, and since $\sum_h \big|d^t_{\theta_{\text{old}}}(h \mid x) - \rho_t(h \mid x)\big| = 2\,\mathrm{TV}(d^t_{\theta_{\text{old}}}, \rho_t)$,

$$\big|\mathbb{E}_{d^t_{\theta_{\text{old}}}}[f_t] - \mathbb{E}_{\rho_t}[f_t]\big| \le \|f_t\|_\infty \sum_h \big|d^t_{\theta_{\text{old}}}(h \mid x) - \rho_t(h \mid x)\big| \le 2B\,\mathrm{TV}\big(d^t_{\theta_{\text{old}}}(\cdot \mid x), \rho_t(\cdot \mid x)\big).$$

For the teacher-reliability term, Jensen's inequality and the definition of $\epsilon^\theta_{T,t}$ give $\big|\mathbb{E}_{\rho_t}[f_t - g_t]\big| \le \mathbb{E}_{\rho_t}[|f_t - g_t|] = \mathbb{E}_{\rho_t}[\epsilon^\theta_{T,t}]$. Applying the triangle inequality, multiplying by $\alpha_t \ge 0$, summing over t, and taking $\mathbb{E}_{x\sim\mathcal{D}}$ establishes (4). □

C Further Method Details

From the bridge to a step-decaying weight. The bridge weight reshapes the collected roll-in $\mathcal{P}_t \approx d^t_T$ toward $\rho^\star_t$. Substituting the bridge (6) gives a weight that is a power of the student-to-teacher occupancy ratio,

$$w_t(x, h_t) \;\propto\; \frac{\rho^\star_t(h_t \mid x)}{\mathcal{P}_t(h_t \mid x)} \;=\; \left(\frac{d^t_{\theta_{\text{old}}}(h_t \mid x)}{d^t_T(h_t \mid x)}\right)^{\gamma_t}.$$

For a teacher-recorded prefix h_t, both policies face the same observations. The environment factors therefore cancel, leaving a cumulative student-to-teacher action likelihood ratio:

$$\widehat{r}_t(x, h_t) = \prod_{s<t} \frac{\pi_{\theta_{\text{old}}}(a_s \mid x, h_s)}{\pi_T(a_s \mid x, h_s)}.$$

This ratio is a genuine per-history quantity – it varies across the histories at a fixed step t – so the weight that realizes the bridge inside the weighted objective (2) is its per-(x, t)-normalized power

$$w_t(x, h_t) = \frac{\widehat{r}_t(x, h_t)^{\,\gamma_t}}{\mathbb{E}_{H_t\sim\mathcal{P}_t(\cdot\mid x)}[\widehat{r}_t(x, H_t)^{\,\gamma_t}]}, \qquad \mathbb{E}_{H_t\sim\mathcal{P}_t}[w_t] = 1, \tag{7}$$

which reshapes the pooled roll-in into the effective per-step distribution $\rho_t = w_t\,\mathcal{P}_t$ exactly as the general framework prescribes, upweighting the histories the current student is relatively more likely to have produced. Because $\widehat{r}_t$ varies within a step this normalization is non-vacuous, so Eq. (7) is ReOPD's exact, density-ratio realization of the bridge; it is computable from the student and teacher log-probabilities on the recorded prefix, at the price of a high-variance per-history ratio.

Each factor compares the student's and the teacher's probability of the teacher's own recorded action a_s: since the teacher selected a_s, the ratio $\pi_{\theta_{\text{old}}}(a_s)/\pi_T(a_s)$ is typically below one, so the product *shrinks* as the prefix lengthens. Equivalently, $\log \widehat{r}_t$ is a cumulative sum that becomes more negative with depth: the further the supervised position sits in the trajectory, the less likely the current student would have followed the same teacher prefix, and the larger the surrogate mismatch between the pooled prefix and the student-relevant history distribution. In other words, the reliability weight is, to first order, a monotone *step-decaying* function of the position index.

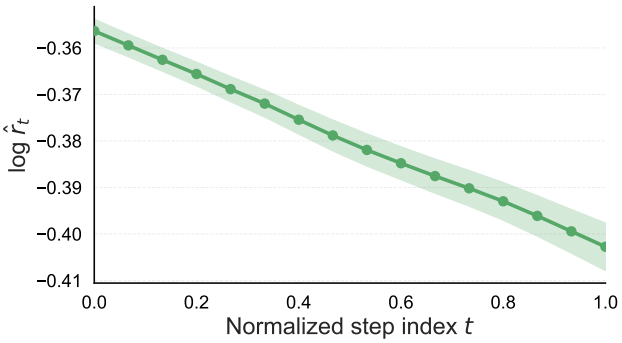


Figure 5: **Step index is a strong proxy for the likelihood-ratio weight.** For each supervised position we compute the per-position weight $\widehat{r}_t = \prod_{s<t} \pi_{\theta_{\text{old}}}(a_s)/\pi_T(a_s)$ – the student-to-teacher likelihood ratio along the teacher prefix – and plot it against the step index t. The empirical weight decays with depth, because each factor compares the student's and teacher's probability of the teacher's own recorded action and is typically below one, so the product shrinks as the prefix lengthens. This makes step index a cheap proxy for the expensive prefix-level ratio and motivates replacing the explicit ratio with the one-parameter step-decay schedule $\omega(t) = \kappa^t$ introduced below.

The ratio also decays with step index along recorded teacher prefixes (Figure 5). This motivates using depth as a stable proxy rather than estimating a high-variance ratio for each prefix.

Concretely, the surrogate keeps the monotone-decreasing *shape* of the exact weight (7) while dropping the noisy per-history factor $\widehat{r}_t$; both the shape and the right steepness follow from the exact weight itself. Writing the average per-step teacher–student gap along the prefix as

$$\bar{c}(x) = \mathbb{E}_{a_s \sim \pi_T}\left[\log \frac{\pi_T(a_s \mid x, h_s)}{\pi_{\theta_{\text{old}}}(a_s \mid x, h_s)}\right] \;\geq\; 0 \qquad \text{(an average per-step } D_{\text{KL}}(\pi_T \,\|\, \pi_{\theta_{\text{old}}})\text{)},$$

we have $\log \widehat{r}_t \approx -(t-1)\,\bar{c}$, so the exact weight's depth profile is geometric:

$$\widehat{r}_t^{\,\gamma_t} \;\approx\; \kappa^{\,t} \text{ (up to a constant)}, \qquad \kappa = \exp(-\gamma_t\,\bar{c}) \in (0, 1]. \tag{8}$$

This is the bridge-to-schedule map: the decay *base* κ is not a free prefactor but is pinned by the bridge exponent γ_t (how far ρ_t is pushed toward the student occupancy) and the per-step gap $\bar{c}$ (how fast the teacher pool diverges from the student with depth), and shrinks – steeper decay – when either grows. Collapsing the profile to a single base κ treats the per-step gap as approximately stationary across positions; when it instead grows with depth, κ^t is a first-order (geometric-mean) approximation to the true accumulated ratio, which Figure 5 shows already tracks the measured decay well. ReOPD therefore uses the one-parameter schedule

$$\omega(t; \kappa) = \kappa^{\,t} \qquad \text{(or any monotone decreasing schedule)},$$

with $\kappa \in (0, 1]$ the single steepness knob; the overall scale cancels under normalization, so – unlike a raw power law – there is no separate prefactor, and in particular no scale symbol to collide with the bridge exponent γ_t. This is the surrogate's reliability weight, $w_t = \omega(t; \kappa) = \kappa^{\,t}$. Because it depends only on the position index it is constant across the histories at a fixed t, so it is normalized *over the horizon* – across positions, in proportion to $\kappa^{\,t}$ – rather than per-(x, t); a per-(x, t) normalization would send a step-constant weight to $w_t \equiv 1$ and erase the decay. This makes the trade-off between the two forms explicit: both feed the same weighted objective (2) through the effective distribution $\rho_t \propto w_t\,\mathcal{P}_t$, but the exact weight (7) varies *within* each step (per-(x, t) normalization – a genuine density ratio that reshapes histories at a fixed t), whereas the step-decay surrogate varies *across* steps (horizon normalization – constant within a step) and so reallocates mass toward the early, low-shift positions. They differ only in the axis along which w_t carries the reliability signal. The surrogate trades exactness for a single interpretable knob: $\kappa = 1$ recovers uniform weighting across positions (no decay), and smaller κ concentrates supervision on the early, more reliable steps. Both forms point the same way, since $\widehat{r}_t$ decays with depth; the surrogate is the default we sweep in Section 4, realized by sampling positions in proportion to it (equivalently, as a loss weight), as detailed next. More generally the weight can be made data-dependent – e.g. derived from the student's own sampling probabilities at the evaluated step.

From the objective to sampling. The design target is a *distribution*, not a weight: objective (3) is the plain (unweighted) loss in expectation under $\rho_t^\star$. Any unbiased Monte-Carlo estimate of that expectation is a valid implementation, and there are two natural ones. *(i) Sampling (direct).* Estimate (3) by drawing histories from $\rho_t^\star$ and averaging the *unweighted* loss. Since the pool only provides samples from $\mathcal{P}_t$, we draw positions with $p_t \propto w_t \mathcal{P}_t = \rho_t^\star$ – i.e. resample the pooled positions in proportion to the step-decay schedule – and apply the loss with no reweighting. This is the direct estimator of (3): the schedule appears only in *which* positions are trained, matching how the offline pool is streamed and touching only a subset of positions per step. *(ii) Weighting (importance).* Keep the pooled samples from $\mathcal{P}_t$ and instead multiply the loss by w_t. The importance identity $\mathbb{E}_{H_t \sim \mathcal{P}_t}[w_t\, \ell] = \mathbb{E}_{H_t \sim \rho_t^\star}[\ell]$ makes this the importance-weighted estimator of the *same* expectation, at a compute cost proportional to enumerating every position. Sampling and weighting estimate the same objective (3) without bias. We use direct sampling throughout.

Relation to RL and distillation. ReOPD sits between reinforcement learning and distillation, and the bridge parameter γ_t is precisely the slider between them. Like on-policy RL, the student acts at the supervised step and is trained on its own actions, so the signal is relevant to what the student will do; like distillation, the supervision is the teacher's full per-step conditional $\pi_T(\cdot \mid x, h_t)$ rather than a scalar return. This is the sense in which on-policy distillation provides a *dense* improvement signal where RL provides a sparse one (Lu & Thinking Machines Lab, 2025): RL conveys a few bits per episode, whereas matching a distribution supervises every token. In our occupancy language, $\gamma_t \to 1$ places the effective prefix distribution at the student occupancy $d^t_{\theta_{\text{old}}}$ – the regime on-policy methods target – while $\gamma_t \to 0$ places it at the teacher occupancy d^t_T – ordinary teacher-forced distillation – and intermediate values interpolate.

This should not be mistaken for off-policy or reward-weighted RL. We do not estimate returns or advantages, apply no importance-sampling correction to a value function, and never optimize a scalar reward; the target is a fixed teacher distribution and the weight w_t encodes the reliability and student-relevance of the replayed prefix, not a policy-gradient ratio. Nor do we require environment interaction: the "on-policy" part is the student's action at a single step on a replayed prefix, not a fresh rollout. ReOPD is thus best read as distillation with an on-policy supervised step and a reliability-aware prefix distribution – inheriting the relevance of RL-style on-policy data and the density of distillation, while avoiding both the sparsity of reward and the cost of online rollouts.

D Experimental Setup

Models. We use Qwen3-family models (Yang et al., 2025) as both teachers and students. To vary the teacher–student capability gap, we distill from teachers of three scales – Qwen3-4B-Instruct-2507, Qwen3-8B, and Qwen3-30B-A3B-Instruct-2507 – into a Qwen3-4B-Instruct-2507 student; we additionally use Qwen3-8B as a student under the Qwen3-30B-A3B-Instruct-2507 teacher.

All models first receive supervised fine-tuning (SFT) on trajectories collected by a stronger model to teach tool use and the required output format. Without this cold start, the model often ignores the Python tool on math tasks. We then train the teacher with GRPO and distill into a student initialized from the cold-start checkpoint, as in the practical sequence of teacher RL followed by distillation.

By default, the teacher prefix pool is the set of agentic trajectories produced by the teacher while being trained with on-policy RL (GRPO). In this way, we can reuse the trajectories from the training of the teacher model without any extra collection cost (Section 3).

Datasets & environments. We train and evaluate in two agentic tasks, both following the same three-stage protocol: an SFT cold start on 2K teacher trajectories (for both teacher and student), GRPO (Shao et al., 2024) training of the teacher, and student distillation (OPD or ReOPD) on that same prompt set. *(i) Mathematical reasoning (Python-tool environment).* Following ReTool (Feng et al., 2025), cold-start traces are taken from ReTool and the 6.4K prompts for GRPO and distillation are taken from the DAPO training set (Yu et al., 2025). We evaluate on six competition benchmarks – AIME24 (Art of Problem Solving Wiki, 2024), AIME25 (Art of Problem Solving Wiki, 2025), AMC23 (Yang et al., 2024), Minerva Math (Lewkowycz et al., 2022), OlympiadBench (He et al.,

Table 5: **Training parameters.** GRPO trains only the teacher; ReOPD reuses its rollouts. Search uses a larger GRPO batch to reduce noise.

Hyper-parameter	Cold Start	SFT	GRPO	OPD	ReOPD
batch size	64	64	32×8 (n=8) for Math 128×4 (n=4) for Search	256×1 (n=1)	256×1 (n=1)
lr	5e-5	5e-5	1e-6	1e-6	1e-6
lr scheduler	cosine	cosine	constant	constant	constant
min lr	1e-6	1e-6	-	-	-
warmup ratio	0.1	0.1	-	-	-
epoch / steps	3 epochs	5 epochs	200 steps	200 steps	200 steps
clip ($\epsilon_{low}, \epsilon_{high}$)	-	-	(0.2, 0.28)	-	-
temperature	-	-	1.0	1.0	1.0
max_gen_tokens	-	-	8192 for Math	8192 for Math	8192 for Math
	-	-	4196 for Search	4196 for Search	4196 for Search
	-	-	-	8192 for multi-envs	8192 for multi-envs
max number of tool calls	-	-	16 for Math	16 for Math	16 for Math
	-	-	4 for Search	4 for Search	4 for Search
	-	-	-	16 for multi-envs	16 for multi-envs
concurrency of tool calls	-	-	32	32	-
GPU mem for retriver	-	-	80GB	80GB	-

Table 6: **Evaluation hyper-parameters for two environments.**

Hyper-parameter	Math with Python	Search
temperature	1.0	1.0
max_gen_tokens	16384	8192
accuracy avg@n	$n = 8$ (AIME24, AIME25, AMC23) or $n = 4$ (the rest)	$n = 1$
max number of tool calls	16	4

2024), and MATH500 (Lightman et al., 2023; Hendrycks et al., 2021). *(ii) Search (retrieval environment).* Following Search-R1 (Jin et al., 2025), prompts come from the merged NQ (Kwiatkowski et al., 2019) and HotpotQA (Yang et al., 2018) training sets: we randomly sample 2K prompts for cold-start trajectory generation by Qwen3-30B-A3B-Instruct-2507, and another 6.5K prompts for teacher GRPO and student distillation. For retrieval, we use the 2018 Wikipedia dump (Karpukhin et al., 2020) as the knowledge source and E5 (Wang et al., 2022) as the retriever, with top-3 retrieved passages. We evaluate on seven QA benchmarks – general QA (NQ, TriviaQA (Joshi et al., 2017), PopQA (Mallen et al., 2023)) and multi-hop QA (HotpotQA, 2WikiMultiHopQA (Ho et al., 2020), MuSiQue (Trivedi et al., 2022), Bamboogle (Press et al., 2023)) – with NQ and HotpotQA in-domain (†) and the remaining five out-of-domain (*). The statistics for the evaluation sets is listed in Table 7

For multi-environment training, one student learns from shuffled data from both tasks for SFT and OPD/ReOPD and is evaluated on all benchmarks. The teachers remain domain-specific.

Baselines. We compare ReOPD against the following methods. *Base*: the untrained model. *Cold Start*: the model is trained with SFT for the cold start stage. This is also the initialized model for the following methods and ReOPD. *SFT*: the model is trained on the trajectories collected from the teacher model (same as ReOPD) with SFT. *OPD*: fully student-on-policy distillation, where prefixes are re-rolled by the current student through the live environment and the teacher supplies the per-step target. *GRPO*: the teacher's own RL result, shown for reference as a soft upper bound at the teacher's scale, not a competing student.

Implementation details. Tables 5 and 6 list the training and evaluation settings. We train RL for 200 steps; ReOPD and OPD use the same teacher target, batch size, and number of updates, differing only in prefix distribution. Unless noted, ReOPD samples with $\kappa = 0.6$ (Section 3). Following Feng et al. (2025) and Jin et al. (2025), we report macro averages for math and micro averages for search. Runs use 8 H100 GPUs and Slime (Zhu et al., 2025); ReOPD takes under 3 hours.

Table 7: **The statistics of evaluation sets.**

Math task	AIME24	AIME25	AMC23	Minerva	Olympiad	MATH500	Sum
Amount	30	30	40	272	675	500	1547
Search tasks	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	Musique	Bamboogle	Sum
Amount	3610	11313	14267	7405	12576	2417	125	51713

E Additional Experiments and Analysis

E.1 Joint Training and Ablations

Figure 6 shows joint training from separate teacher rollouts; Table 3 reports accuracy.

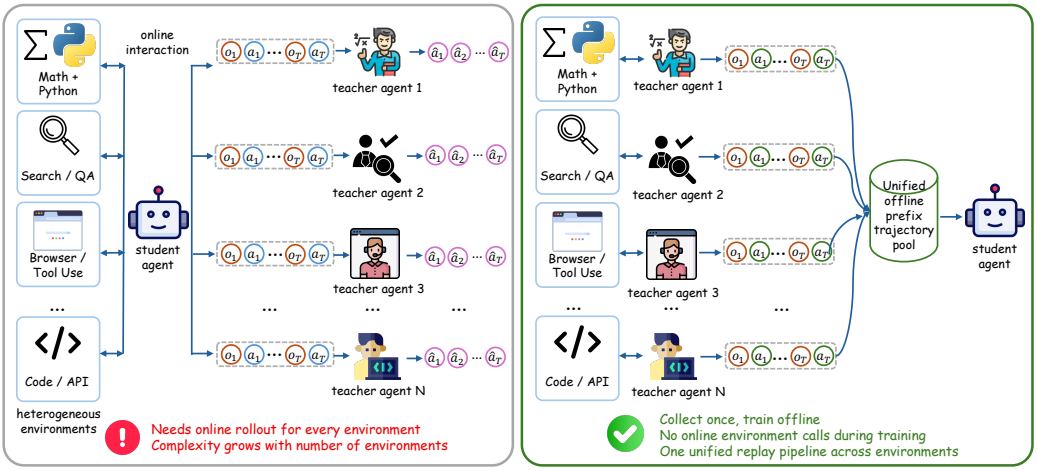


Figure 6: **Training a shared student agent on multiple heterogeneous environments with OPD and ReOPD. Left:** With an increasing number of environments, the operational complexity grows for OPD due to the heavy deployment of the environments. **Right:** ReOPD doesn't require the deployment of all environments at the same time. The teacher's trajectories can be collected separately for different environments, and then be merged into a unified pool for the later training of the student agent without any online environment.

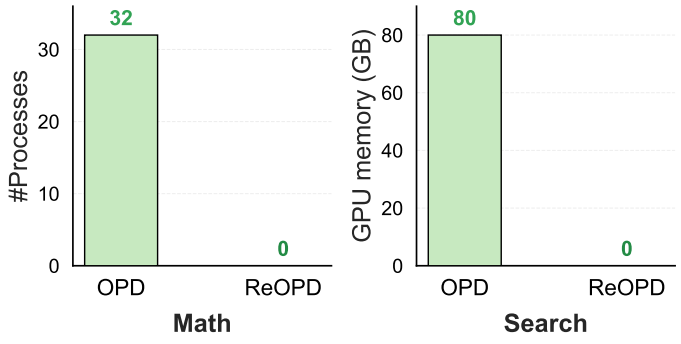


Figure 7: **The resources used by different environments.** For math, the processes are utilized for the execution of Python code in parallel. For search, the GPU memory is utilized for deploying the embedding model and the storage of the Wikipedia embeddings.

E.2 Analysis and Discussion

Validating the step-level schedule. Figure 3 tests whether the depth proxy of Section 3 actually matters for final performance. Because the prefix is teacher-forced and only the current step is student-on-policy, the two-sided shift is smallest at early steps and grows with depth. Both panels use the sampling implementation: varying which chunk is drawn (Fig. 3a) performs best when it draws more early chunks, and varying the decay steepness κ (Fig. 3b) shows the same trend as late steps are sampled less. The agreement between the two knobs corroborates the step-decay schedule of Section 3; the improvement from favoring early, low-shift steps shows that the gain comes from reliability-aware prefix selection rather than simply using more data.

Two regimes in practice. The tables reveal the two regimes anticipated by our analysis. On search and multi-hop QA, where the teachers stay reliable on student-induced histories, the student-occupancy shift dominates: student-on-policy OPD is already strong, and ReOPD – whose teacher-anchored roll-in is already student-relevant here, since the two occupancies nearly coincide – essentially matches it (e.g. a 40.5 vs. 40.6 average under the single-environment setting). On mathematical reasoning the picture changes as the teacher–student gap widens. With stronger teachers (Qwen3-8B and Qwen3-30B-A3B), student roll-ins drift into histories where the teacher's trajectory is the more trustworthy signal, so the teacher-reliability shift dominates; ReOPD's teacher-anchored roll-in sidesteps these unreliable histories and attains its largest gains, improving AIME24 from 28.3 to 36.7 under the Qwen3-8B teacher while lifting the 4B-student math average in every teacher setting ($55.1 \rightarrow 57.2$, $51.0 \rightarrow 53.7$, $51.1 \rightarrow 52.5$). Which roll-in is preferable thus depends on the regime – the teacher-anchored pool wins when the gap is large and ties student-on-policy OPD when the teacher is already reliable, rather than committing to a fixed student-on-policy roll-in. Our analysis prescribes a steeper decay as the teacher–student gap grows (Section 3); notably, these results obtain the regime-appropriate behavior with a single fixed $\kappa = 0.6$ across tasks rather than per-task tuning, and gap-adaptive schedules are a natural extension.

Prefix source: reliability, not raw capability. Step index is only a proxy; what it ultimately stands in for is support overlap between the student-relevant histories and the teacher's reliable region. Figure 4 isolates this by varying *where prefixes come from* while holding the distillation target fixed at the teacher. One might expect prefixes from a larger or stronger generator to help, since such models make fewer mistakes. We observe the opposite: prefixes drawn from the *teacher's own model* are best, and substituting a larger or stronger prefix generator degrades the distilled student. This is a direct test of the teacher-reliability shift. The teacher's recorded conditional is a reliable improvement signal only on the histories the teacher itself is likely to visit; prefixes from a different, stronger model fall outside this support, so querying the teacher there yields a less trustworthy target. The right notion for prefix selection is therefore *reliability with respect to the teacher*, not the standalone capability of the prefix generator – precisely the quantity ReOPD's reliability-aware prefix selection is designed to respect.

RL-collected pool matches a stationary teacher pool. A natural concern is that the RL-collected pool is *not* stationary: it mixes rollouts from weaker early checkpoints with the final teacher, so it need not match a pool drawn purely from the converged teacher. Table 4 tests this by fixing the distillation target to the final teacher and varying only the prefix pool. The two pools yield nearly identical students, confirming that the checkpoint-to-checkpoint drift in RL rollouts is within the teacher's reliable support and that no dedicated collection is needed – the free by-product suffices.

Teacher's prefix resampling. When RL trajectories are unavailable, we can collect teacher rollouts once for all prompts and reuse them as prefixes. Table 4 compares this stationary pool with teacher RL trajectories. Figure 8 includes one-time collection in the OPD comparison. ReOPD remains more than $2\times$ faster per rollout, and student training needs no live environment.

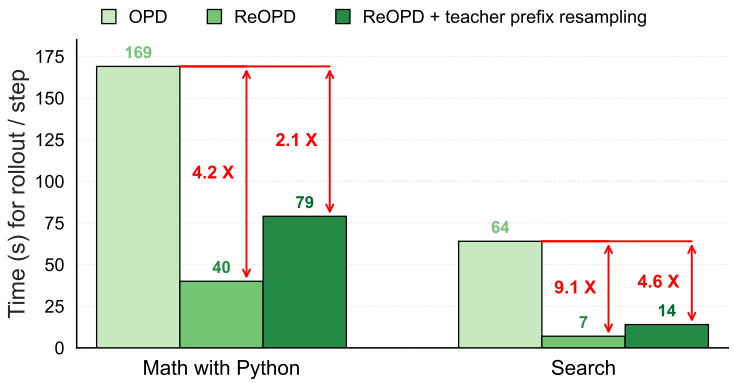


Figure 8: **Rollout time when taking the teacher's rollout into account.** When the teacher's trajectories from RL are not available, one could use the teacher to sample the trajectories for all prompts at once. We take this time into account for the comparison. For this setting, the environment is only online for the teacher's rollout.

F Limitations and Future Work

Our analysis assumes access to a pre-collected pool of teacher trajectories with recorded observations. In our experiments this comes for free from the teacher's own RL rollouts, but more generally the pool's coverage and quality bound what the student can learn. The reliability surrogate we adopt reduces, to first order, to a step-decaying weight, which is effective but coarse: it captures *how deep* a prefix is rather than directly measuring *how far* a given prefix lies from the overlap between student-relevant histories and the teacher's reliable support, and a learned or data-dependent reliability estimate may improve on the schedule. Finally, our two-sided bound treats the teacher reliability error through teacher support as a surrogate; tighter, directly estimable notions of teacher reliability – and schedules or weights that adapt to them online – are a natural next step.