RETROSPECTION MAKES A BETTER LEARNER: RETROSPECTIVE ON-POLICY SELF-DISTILLATION FOR REASONING MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy post-training methods for reasoning models share a common learning loop where the model *attempts* a problem, *observes* corrective information such as a verifier's reward or a reference solution, and *updates* itself where the attempt and the observation diverge. These methods differ in what is observed and how the update is made, but all produce the attempt the same way, where the model simply responds to the question with no earlier attempt to build on. A human, by contrast, re-examines a first attempt and tries again before consulting a solution, and learns more from the solution as a result. Bringing this behavior to a model means training the attempt stage itself, under two requirements set by its position in the loop: 1) no external information, such as a ground-truth answer or reference solution, since attempting precedes observing, and 2) a single generation pass, since we need a policy that attempts in one pass, in the learning that follows and at inference. We introduce **Retrospective On-Policy Self-Distillation (RSD)**, which meets both requirements with *retrospection*. The model conditioned on its own prior attempt serves as the teacher, and its behavior is distilled into a single generation pass. We show that retrospection preserves single-attempt accuracy while expanding pass@k coverage. Our central finding is that an RSD-trained policy is a better learner, so that on-policy learning schemes initialized from it achieve consistently higher performance across benchmarks. Code will be released.

1 INTRODUCTION

Despite their differences, on-policy post-training methods for reasoning models follow a common learning loop: the model *attempts* a problem, *observes* corrective information, and *updates* itself where the attempt and the observation diverge (Figure 1-(a) & Table 1). For instance, Reinforcement Learning with Verifiable Rewards (RLVR) instantiates the loop with a verifier's reward as the observation and a policy gradient as the update (Jaech et al., 2024; Guo et al., 2025; Shao et al., 2024), and On-Policy Self-Distillation (OPSD) instantiates it with a reference solution as the observation and per-token distillation from the model's own solution-conditioned distribution as the update (Zhao et al., 2026; Hübotter et al., 2026). These methods differ in what is observed and how the update is made, and progress has concentrated on these two stages (Yoon et al., 2025; Agarwal et al., 2024; Gu et al., 2024; Yoon et al., 2026; Shan et al., 2026).

Human learning, however, suggests that how one attempts affects what can be learned in the subsequent observe and update stages. Research in cognitive science, for instance, finds that learners who generate and revise their own solutions before being shown one learn more than those given the solution directly (Kapur, 2014; Lyster, 2004). We therefore ask whether a model whose attempts are improved in the same way becomes a better learner as well, so that the subsequent stages of the loop, such as RLVR or OPSD, learn more from it.

The position of the attempt stage in the loop places two requirements on how this can be done. First, the attempt must be improved without external information, such as a ground-truth answer or a reference solution, since attempting precedes observing. Second, the improvement must be realized in a single generation pass, since both the learning that follows and inference with the trained model draw attempts in one pass from the bare query (the query alone, with no prior attempt in context);

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.

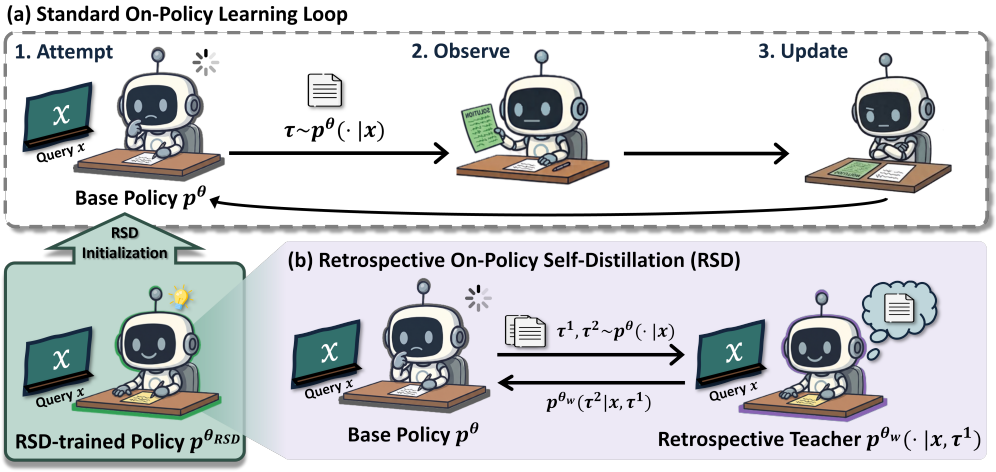

Figure 1: **RSD improves the attempt stage of the learning loop. (a)** On-policy post-training shares a common loop: ① an **attempt** $\tau \sim p^{\theta}(\cdot \mid x)$ is drawn, ② corrective information is **observed**, such as a reward $r(\tau)$ or a reference solution τ^*, and ③ the policy is **updated** where the two diverge. Existing frameworks improve the signal and the update, but the attempt always comes from the base policy, with no earlier attempt to build on. **(b)** Just as a human re-examines a first attempt and tries again before consulting a solution, the model is conditioned on its own prior attempt $\tau^1 \sim p^{\theta}(\cdot \mid x)$, which gives the retrospective teacher $p^{\theta_w}(\cdot \mid x, \tau^1)$ (θ_w is a briefly warmed-up copy of θ, Section 4). RSD distills this teacher into the single-pass policy $p^{\theta}(\cdot \mid x)$ along the policy's own attempts τ^2, so that its attempt then has retrospective quality without a prior attempt in context. We show that a loop initialized from the resulting policy $p^{\theta_{\mathrm{RSD}}}$ learns more than one initialized from the base policy.

the improvement must therefore be a property of the policy rather than an inference-time procedure. For a language model, generating and revising its own solution has a direct analogue that satisfies the first requirement: conditioning the model on its own prior attempt, which we call *retrospection* (Figure 1-(b)). The second requires making retrospection a property of the unconditioned policy.

To this end, we propose **Retrospective On-Policy Self-Distillation (RSD)**, in which the model conditioned on its own prior attempt serves as the teacher and the unconditioned model, as the student, distills it on its own rollouts. The unconditioned policy thereby acquires the behavior of its retrospective counterpart, and a single generation pass from the bare query produces attempts of retrospective quality. **In summary, our contributions are:**

- **Retrospection Expands Coverage.** Conditioning the model on its own prior attempt expands pass@k coverage without reducing single-attempt accuracy, and the gain in coverage comes from problems the base policy solves rarely or never (Section 3). Coverage matters for the loop that follows, since on-policy learning can reinforce only the solutions that the attempt distribution reaches (Yue et al., 2025; Gandhi et al., 2025).
- **Length Collapse and Teacher Warmup.** Since the loop needs a single-pass policy, the conditioned policy must be distilled into the unconditioned one. However, distilling it directly leaves the student below its starting point. We trace the failure to a *length collapse*: with an attempt in its context, the conditioned policy omits the reasoning steps that the context makes redundant, and distillation transfers the truncated form to a student that lacks the context. We resolve it with a brief, role-reversed *teacher warmup*, in which the unconditioned model teaches the conditioned one to reason at full length while the prior attempt stays in its context (Section 4).
- **Retrospection Makes a Better Learner.** The RSD-trained policy retains the coverage gain of retrospection without a prior attempt at inference. Given the query alone, a single generation pass reaches the pass@k of the conditioned policy (Section 6.1). Our main result is that the on-policy learning loop learns more when initialized from the RSD-trained policy. On Qwen3-4B and Qwen3-8B, RLVR (GRPO) (Shao et al., 2024) and self-distilled RLVR (RLRT) (Kim et al., 2026) started from it gain up to 9.5 points in average pass@1 over five benchmarks relative to the same runs started from the base policy, with the training configuration otherwise unchanged (Section 6.2). Additionally, we show that for GRPO this advantage persists even when the base run is given the FLOPs consumed by RSD (Section 6.3).

Table 1: **On-policy post-training as an ① Attempt, ② Observe, ③ Update loop.** Frameworks differ in what is observed and how the policy is updated, but the attempt is drawn in the same way in each and is not itself trained. RSD trains the attempt stage and leaves the other two unchanged, so that the policy becomes a better learner within existing frameworks.

Framework	① Attempt	② Observe	③ Update
RLVR (GRPO) (Shao et al., 2024)	$\{\tau^i\}_{i=1}^G \sim p^\theta(\cdot \mid x)$	verifier reward $r(\tau^i)$	group-normalized policy gradient
OPD (Agarwal et al., 2024)	$\tau \sim p^\theta(\cdot \mid x)$	teacher distribution q_T	token-level divergence to q_T
OPSD (Zhao et al., 2026)	$\tau \sim p^\theta(\cdot \mid x)$	reference solution τ^*	token-level divergence to $p^\theta(\cdot \mid x, \tau^*)$
Self-distilled RLVR (RLRT) (Kim et al., 2026)	$\{\tau^i\}_{i=1}^G \sim p^\theta(\cdot \mid x)$	$r(\tau^i)$ and τ^*	policy gradient reweighted by $p^\theta(\cdot \mid x, \tau^*)$
RSD (ours)	**trained**	*unchanged: compatible with the frameworks above*	

2 Problem Formulation and Preliminaries

2.1 Problem Formulation: Improving the Attempt Stage

Given a query x, a policy p^θ generates a reasoning trajectory $\tau = (\tau_1, \ldots, \tau_T)$, a chain of thought followed by a final answer. Despite their differing objectives, on-policy post-training frameworks share the loop of Figure 1-(a), summarized in Table 1: ① an attempt $\tau \sim p^\theta(\cdot \mid x)$ is drawn, ② corrective information is observed, and ③ the policy is updated where the two diverge. Prior work has concentrated on the latter two stages for richer observations and better update rules, while the attempt is drawn in the same way in every framework and is not itself trained.

This paper specifically targets the attempt stage. In human learning, a learner who re-examines a first attempt and tries again before consulting a solution learns more from that solution than one who is given it directly (Kapur, 2014; Lyster, 2004). We ask whether the attempt of a model can be improved in the same way, and whether the loop then learns more from it. Any such improvement must respect *two requirements* that follow from the attempt's position in the loop. **(i) No external information.** Attempting precedes observing, so labels, rewards, or verifiers are not available; whatever improves the attempt must come from the model alone. **(ii) Single generation pass.** Both the learning that follows and inference with the trained model should draw attempts in one generation pass from the query x, so the improvement must be internalized into $p^\theta(\cdot \mid x)$ rather than implemented as an inference-time procedure.

Let θ_0 denote the parameters of the initial policy. *Retrospection*, conditioning the model on its own prior attempt, meets the first requirement by construction: a prior attempt $\tau^1 \sim p^{\theta_0}(\cdot \mid x)$ is sampled and placed in the model's context, giving the conditioned policy $p^{\theta_0}(\cdot \mid x, \tau^1)$, which re-examines the model's own work before any solution is seen. Meeting the second requires internalizing this conditioned behavior into the unconditioned policy: we seek updated parameters θ for which $p^\theta(\cdot \mid x) \approx p^{\theta_0}(\cdot \mid x, \tau^1)$ for $\tau^1 \sim p^\theta(\cdot \mid x)$, so that a single pass from the bare query reproduces the retrospective behavior of the initial policy. *Two questions* therefore decide whether this is viable: whether $p^{\theta_0}(\cdot \mid x, \tau^1)$ is a better policy than $p^{\theta_0}(\cdot \mid x)$, and in what sense; and, if so, how its advantage can be transferred into $p^\theta(\cdot \mid x)$. We address the first question in Section 3 and the second in Section 4.

2.2 Preliminaries: On-Policy Self-Distillation (OPSD)

Before turning to these questions, we review on-policy self-distillation (OPSD), on which RSD builds. It transfers a conditioned distribution of a model into its unconditioned counterpart using only the model's own rollouts, which is precisely the transfer the second question requires.

On-Policy Distillation (OPD). OPD (Gu et al., 2024; Agarwal et al., 2024) trains a student policy p^θ to match a teacher distribution q_T through token-level supervision on the student's own rollouts. For a trajectory τ and two autoregressive distributions p and q, let

$$\ell(\tau;\, p,\, q) \;\triangleq\; \frac{1}{|\tau|} \sum_{t=1}^{|\tau|} D_{\mathrm{JSD}(\beta)}\big(\, q(\cdot \mid \tau_{<t}) \,\big\|\, p(\cdot \mid \tau_{<t})\big) \tag{1}$$

denote the mean token-level divergence along τ, where $D_{\mathrm{JSD}(\beta)}$ is the generalized Jensen–Shannon divergence (Agarwal et al., 2024), which interpolates between the forward KL divergence

$D_{\mathrm{KL}}(q \,\|\, p)$ as $\beta \to 0$ and the reverse KL divergence $D_{\mathrm{KL}}(p \,\|\, q)$ as $\beta \to 1$, and $p(\cdot \mid \tau_{<t})$ is the next-token distribution that p assigns at step t given the prefix $\tau_{<t}$ and any context p is itself conditioned on. OPD minimizes

$$\mathcal{L}_{\mathrm{OPD}} \triangleq \mathbb{E}_{\tau \sim p^{\theta}(\cdot|x)} \Big[\ell\big(\tau;\ p^{\theta}(\cdot \mid x),\ q_T\big) \Big]. \tag{2}$$

Sampling the trajectories from the student, rather than from the teacher, places the supervision on the states the student visits at inference, which removes the mismatch between training and inference distributions of distillation on teacher-generated text (Agarwal et al., 2024). OPD has attracted considerable attention as a post-training stage in both the language (Yang et al., 2025; Jin et al., 2026; Yu et al., 2026b; Ge et al., 2026) and vision-language domains (Yoon et al., 2026; Liu et al., 2026).

On-Policy Self-Distillation (OPSD). OPSD (Zhao et al., 2026; Hübotter et al., 2026) removes the external teacher; the same model, given privileged information in its context, assigns sharper next-token distributions along a rollout and thereby serves as its own teacher. Let $\mathrm{sg}[\cdot]$ denote the stop-gradient operator, so that no gradient flows through the teacher side of the divergence. Given a reference solution τ^* for query x (i.e., a correct solution either supplied externally or selected from the model's own attempts by a verifier), the teacher is $p^{\theta}(\cdot \mid x, \tau^*)$, and OPSD minimizes

$$\mathcal{L}_{\mathrm{OPSD}} \triangleq \mathbb{E}_{\tau \sim p^{\theta}(\cdot|x)} \Big[\ell\big(\tau;\ p^{\theta}(\cdot \mid x),\ \mathrm{sg}[p^{\theta}(\cdot \mid x, \tau^*)]\big) \Big], \tag{3}$$

which is Eq. 2 with the model's own solution-conditioned distribution in place of q_T. The supervision is concentrated where the rollout diverges from the reference solution, where the two distributions differ most.

OPSD thus internalizes a conditioned distribution of the model, $p^{\theta}(\cdot \mid x, \tau^*)$, into its unconditioned policy $p^{\theta}(\cdot \mid x)$, which is the transfer the second requirement calls for. RSD adapts this objective to the attempt stage by placing the prior attempt τ^1 in the teacher's context in place of the reference solution τ^*. RSD *borrows only the distillation mechanism of* OPSD*, not its role in the loop.* OPSD requires a reference solution and therefore instantiates the observe and update stages, whereas RSD requires none and trains the attempt stage. The two are complementary rather than competing, and a policy trained with RSD in fact learns more from self-distilled RLVR, which carries the signal of OPSD (Section 6.2).

The substitution is not trivial, however, because a teacher shown the reference solution is better informed than the student by construction, whereas a teacher shown a possibly wrong attempt carries no such guarantee. Whether $p^{\theta}(\cdot \mid x, \tau^1)$ is a better teacher at all, and if so in what sense, is therefore the first question to settle.

3 Benefits of Retrospection and the Challenge of Distilling It

Analysis Setup. We use Qwen3-8B (Yang et al., 2025) and evaluate on AIME 2024–2026. For each query x, we sample a prior attempt $\tau^1 \sim p^{\theta}(\cdot \mid x)$, then evaluation trajectories from the conditioned policy $p^{\theta}(\cdot \mid x, \tau^1)$, and compare against trajectories from the base policy $p^{\theta}(\cdot \mid x)$, with the prior attempt in context as the *only* difference. We draw 1,024 trajectories per problem from each policy and compute each problem's solve rate and pass@k, the probability of at least one correct answer within k sampled attempts, using the unbiased estimator of Chen et al. (2021).

Retrospection expands coverage, and the gain comes from the hardest problems. In Figure 2-(a), the conditioned policy's pass@k pulls away from the base policy's as k grows, reaching **+4.6** points at k=512. Since pass@k at large k measures *coverage*, the fraction of problems for which a policy produces at least one correct solution (Brown et al., 2024; Yue et al., 2025), the gap means that the conditioned policy solves problems the base policy does not. Figure 2-(b) shows where the added problems come from by grouping problems according to the base policy's solve rate. The gain is concentrated in the problems the base policy never solves, without compromising performance on the problems it already solves reliably. Retrospection thus reaches new problems without giving up those already within reach.

Coverage matters for the learning that follows, since on-policy learning learns only from the trajectories that the policy itself samples (Yue et al., 2025; Gandhi et al., 2025), and an attempt stage that

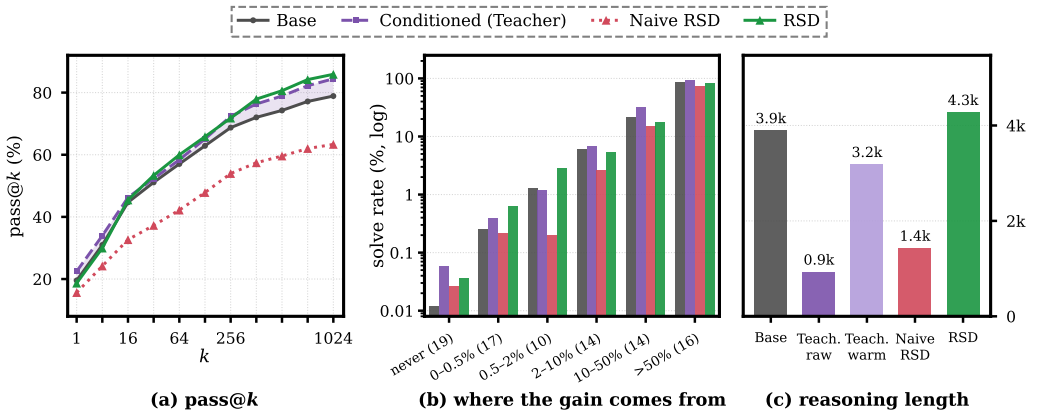

Figure 2: **Retrospection expands coverage, and RSD transfers it into a single generation pass.** Qwen3-8B on AIME 2024–2026. *Base* is the unconditioned policy and *Conditioned* is the same model with its own prior attempt in context. The two students are distilled from the conditioned policy, either directly (*naive* RSD) or after the teacher warmup of Section 4 (RSD); unlike the conditioned policy, both are evaluated without a prior attempt in context. **(a)** pass@k. Conditioning improves coverage at large k. Direct distillation leaves the student below the base policy, whereas RSD matches the conditioned policy. **(b)** Solve rate per problem, averaged within bins of base-policy difficulty (the number of problems in each bin is next to its label). The gain comes from problems the base policy never solves. **(c)** Mean reasoning length. Conditioning shortens the reasoning from 3.9k to 0.9k tokens and the naive student inherits this (1.4k); the warmup restores the teacher to 3.2k, and RSD reasons at full length (4.3k). Qwen3-4B results are in Appendix D.1.

samples correct trajectories on more problems therefore widens what downstream learning can build on (Section 6.2).

Naive distillation transfers the conditioned policy's length collapse rather than its coverage. The natural next step is to distill the conditioned policy directly. Letting θ_0 denote the parameters of the initial policy, we train the student $p^{\theta}(\cdot \mid x)$ on its own rollouts toward the conditioned initial policy $p^{\theta_0}(\cdot \mid x, \tau^1)$. This is Eq. 3 with τ^1 in place of τ^* and the initial policy as the fixed teacher, which we call *naive* RSD. The red curve in Figure 2-(a) shows the result: the student falls *below* the base policy at every k, and Figure 2-(b) shows that the loss falls on the problems the base policy already solves reliably, so that instead of inheriting the conditioned policy's coverage the student loses part of its own. Figure 2-(c) locates the cause in the reasoning length: the base policy averages **3.9**k tokens per problem, the conditioned policy **0.9**k, and the *naive* RSD student trained to imitate it **1.4**k. With an attempt in context, the conditioned policy omits the derivations it treats as established, much as a speaker omits what has already been said (Grice, 1975). For the conditioned policy the omissions are harmless, since the skipped steps are recoverable from τ^1; for the student, which reasons from the bare query x, they are not, and distillation transfers the truncated form without the context that made it harmless. Since the conditioned policy solves more problems than the base policy (Figure 2-(a)), the deficit lies in the surface form of its reasoning rather than in its capability, and a fix must restore its full-length reasoning without erasing the coverage advantage of retrospection. The RSD curves in Figure 2 show the outcome of such a fix, which is introduced in Section 4.1, and Section 6.1 discusses this outcome in detail.

4 Method: Retrospective On-Policy Self-Distillation (RSD)

Section 3 established that the conditioned policy is worth distilling but cannot be distilled as it is. It solves problems that the base policy does not, yet with an attempt in its context its reasoning collapses to about a quarter of its length (Section 3), and a student trained to match it from the bare query inherits the collapsed form without the context that made it harmless. Since the deficit is one of form rather than capability, we repair the teacher before distilling it: the unconditioned policy reasons at full length and can teach the conditioned policy to write out the steps it omits. The prior attempt stays in the conditioned policy's context throughout, since the coverage advantage comes

Algorithm 1 Retrospective On-Policy Self-Distillation (RSD)

Require: Initial policy p^{θ_0}, query distribution $\mathcal{D}$, warmup steps N_{w}, distillation steps N
Stage 1 (Teacher warmup): $\theta \leftarrow \theta_0$; student $p^{\theta}(\cdot \mid x, \tau^1)$, teacher $p^{\theta_0}(\cdot \mid x)$
for $n = 1, \ldots, N_{\text{w}}$ **do**
  Sample $x \sim \mathcal{D}$ and two attempts $\tau^1, \tau^2 \sim p^{\theta}(\cdot \mid x)$ from the bare query
  Update θ by descending $\nabla_\theta \, \ell\big(\tau^2;\ p^{\theta}(\cdot \mid x, \tau^1),\ p^{\theta_0}(\cdot \mid x)\big)$ ▷ Eq. 4
end for
Freeze $\theta_{\text{w}} \leftarrow \theta$ at an early checkpoint, so that the context still matters
Stage 2 (Retrospective distillation): $\theta \leftarrow \theta_0$; student $p^{\theta}(\cdot \mid x)$, teacher $p^{\theta_{\text{w}}}(\cdot \mid x, \tau^1)$
for $n = 1, \ldots, N$ **do**
  Sample $x \sim \mathcal{D}$ and two attempts $\tau^1, \tau^2 \sim p^{\theta}(\cdot \mid x)$ from the bare query
  Update θ by descending $\nabla_\theta \, \ell\big(\tau^2;\ p^{\theta}(\cdot \mid x),\ p^{\theta_{\text{w}}}(\cdot \mid x, \tau^1)\big)$ ▷ Eq. 5
end for
$\theta_{\text{RSD}} \leftarrow \theta$; **return** $p^{\theta_{\text{RSD}}}$

from conditioning on it. RSD therefore consists of two stages. A brief *teacher warmup* restores the full-length reasoning of the conditioned policy (Section 4.1). *Retrospective distillation* then uses the warmed-up conditioned policy as the teacher and transfers it into the unconditioned policy (Section 4.2). Figure 1-(b) and Algorithm 1 summarize the procedure.

4.1 Teacher Warmup: Restoring Full-Length Reasoning

Role-Reversed Objective. We begin with a short distillation phase in which the roles are reversed: the conditioned policy $p^{\theta}(\cdot \mid x, \tau^1)$ becomes the student, and the unconditioned initial policy $p^{\theta_0}(\cdot \mid x)$, with θ_0 kept frozen, becomes its teacher,

$$\mathcal{L}_{\text{warmup}} \triangleq \mathbb{E}_{\tau^1,\, \tau^2 \sim p^{\theta}(\cdot \mid x)} \Big[\ell\big(\tau^2;\ p^{\theta}(\cdot \mid x, \tau^1),\ p^{\theta_0}(\cdot \mid x)\big) \Big]. \tag{4}$$

Trained on full-length rollouts τ^2 of the unconditioned policy while τ^1 remains in its context, the conditioned policy learns to spell out the steps it had been omitting.

Early Stopping. Eq. 4 has a degenerate solution at convergence, where the conditioned policy matches the unconditioned distribution by learning to ignore its context, and the coverage advantage of Section 3 is lost with it. We therefore stop the warmup early, once the conditioned policy's reasoning length has largely recovered (Figure 2-(c)) but before the influence of the context is erased. We freeze this checkpoint (detail in Table 4) as the warmup teacher and denote it by $p^{\theta_{\text{w}}}$.

4.2 The RSD Objective

Given the warmup teacher, RSD proceeds as on-policy self-distillation with a self-generated privilege. For each query x, the student samples a prior attempt τ^1 and a learning trajectory τ^2, and minimizes the token-level divergence to the warmup teacher conditioned on τ^1 along τ^2:

$$\mathcal{L}_{\text{RSD}} \triangleq \mathbb{E}_{\tau^1,\, \tau^2 \sim p^{\theta}(\cdot \mid x)} \Big[\ell\big(\tau^2;\ p^{\theta}(\cdot \mid x),\ p^{\theta_{\text{w}}}(\cdot \mid x, \tau^1)\big) \Big]. \tag{5}$$

Eq. 5 is the OPSD objective (Eq. 3) with the prior attempt τ^1 in place of the reference solution τ^* and the warmup teacher $p^{\theta_{\text{w}}}$ in place of the stop-gradient copy of p^{θ}. Both requirements of Section 2.1 hold by construction. The prior attempt is sampled from the model itself, so (i) no external information enters. Since τ^1 appears only on the teacher side, (ii) the student is always conditioned on the bare query alone. By matching a teacher that has seen a prior attempt, the student internalizes the behavior of a policy that has already retrospected. After training, a single generation pass from the resulting policy, which we denote by $p^{\theta_{\text{RSD}}}$, thus produces attempts of retrospective quality. $p^{\theta_{\text{RSD}}}$ then serves as the initialization of the standard on-policy learning loop of Table 1, whose observe and update stages run unchanged, and Section 6.2 shows that the loop learns more from this retrospective policy than when initialized from the base policy p^{θ_0}.

Table 2: **RSD initialization improves downstream on-policy post-training.** Pass@1 and pass@4 (%) of Qwen3-4B and Qwen3-8B after GRPO and after RLRT, started from the base policy or from the RSD-trained policy under the same configuration. Avg. is the mean over the five benchmarks.

Method	AIME24		AIME25		AIME26		HMMT26		Minerva		Avg.	
	pass@1	pass@4	pass@1	pass@4	pass@1	pass@4	pass@1	pass@4	pass@1	pass@4	pass@1	pass@4
Qwen3-4B												
RLVR (GRPO) (Shao et al., 2024)												
Base + GRPO	31.1	50.6	24.0	39.1	28.3	43.9	18.6	28.9	**61.2**	67.9	32.6	46.1
RSD + GRPO (Ours)	**32.0**	**51.3**	**26.9**	**40.1**	**30.1**	**44.5**	**21.1**	**30.8**	60.9	**69.8**	**34.2**	**47.3**
Δ	+0.9	+0.6	+2.9	+1.0	+1.8	+0.6	+2.5	+1.8	−0.3	+1.9	+1.6	+1.2
Self-distilled RLVR (RLRT) (Kim et al., 2026)												
Base + RLRT	47.0	65.9	40.8	63.5	37.9	57.5	29.1	39.1	62.0	70.8	43.4	59.4
RSD + RLRT (Ours)	**59.1**	**74.6**	**51.7**	**67.9**	**51.4**	**71.8**	**33.2**	**41.6**	**64.7**	**73.5**	**52.0**	**65.9**
Δ	+12.1	+8.7	+10.8	+4.3	+13.5	+14.3	+4.1	+2.5	+2.7	+2.7	+8.6	+6.5
Qwen3-8B												
RLVR (GRPO) (Shao et al., 2024)												
Base + GRPO	39.4	62.8	30.4	44.4	29.3	50.2	22.4	30.5	63.1	70.2	36.9	51.6
RSD + GRPO (Ours)	**49.2**	**71.4**	**39.2**	**57.8**	**42.1**	**61.4**	**29.8**	**36.4**	**65.9**	**73.2**	**45.2**	**60.0**
Δ	+9.8	+8.6	+8.8	+13.4	+12.8	+11.2	+7.4	+5.9	+2.8	+3.1	+8.3	+8.4
Self-distilled RLVR (RLRT) (Kim et al., 2026)												
Base + RLRT	44.7	65.7	40.5	63.9	39.5	58.3	27.8	37.3	65.9	72.9	43.7	59.6
RSD + RLRT (Ours)	**58.2**	**74.2**	**49.3**	**70.5**	**54.4**	**73.7**	**35.4**	**44.5**	**68.7**	**75.3**	**53.2**	**67.6**
Δ	+13.5	+8.5	+8.8	+6.6	+14.9	+15.4	+7.6	+7.2	+2.8	+2.4	+9.5	+8.0

5 Experimental Setup

Models and Training Data. We use Qwen3-4B and Qwen3-8B (Yang et al., 2025) in non-thinking mode (Appendix C) and train on the 17,917 unique problems of DAPO-Math-17k (Yu et al., 2026a). Neither the warmup nor the RSD objective uses a reward, a verifier, or a reference solution. The warmup run uses a learning rate of 1×10^{-6}, and its step-20 checkpoint serves as the teacher. Distillation uses the same learning rate with 128 prompts per step for 320 steps in both the 4B and the 8B experiments, with β=0.5 in Eq. 1 throughout. Appendix B lists all detailed hyperparameters.

Evaluation. For Sections 3 and 6.1 we evaluate on AIME 2024, 2025, and 2026 (30 problems each) with N=1,024 samples per problem at temperature 0.7 and a maximum generation length of 16,384 tokens (full evaluation details in Appendix B.5), and report pass@k with the unbiased estimator of Chen et al. (2021). Section 6.2 tests whether the RSD-trained policy is a better learner than the base policy by training both with two on-policy learning algorithms, **RLVR (GRPO)** (Shao et al., 2024) and **self-distilled RLVR (RLRT)** (Kim et al., 2026), under the same configuration. We add HMMT February 2026 (33 problems) and Minerva-Math (272 problems) (Lewkowycz et al., 2022) and report pass@1 and pass@4 from N=64 samples with a maximum generation length of 32,768 tokens. Section 6.3 controls for the cost of RSD by continuing the base GRPO runs until their FLOPs match those of RSD and GRPO combined. Appendix D.2 extends the comparison to **OPD** (Agarwal et al., 2024) from a larger external teacher, and Appendix D.3 compares RSD against an **OPSD** (Zhao et al., 2026) initialization.

6 Experimental Results

6.1 RSD Inherits the Conditioned Policy's Coverage in a Single Pass

The green curve in Figure 2-(a) is the RSD-trained policy $p^{\theta_{\text{RSD}}}$, evaluated from the bare query with no prior attempt in context. It matches the base policy at small k and rises above it as k grows, reaching **+6.34** points at k=512 on Qwen3-8B and **+6.30** on Qwen3-4B (Appendix D.1), and it tracks the conditioned policy it was distilled from across the range of k. Figure 2-(b) shows that the student's gain falls in the same bins as the conditioned policy's, on the problems the base policy solves rarely or never, while Figure 2-(c) shows that it reasons at full length (**4.3**k tokens against **3.9**k for the base policy) rather than in the collapsed form that the *naive* student inherited. The single-pass policy thus internalizes the coverage of retrospection.

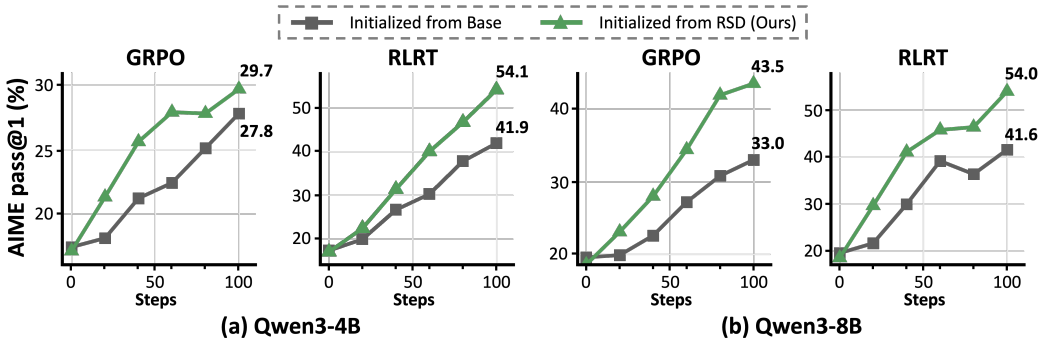

Figure 3: **Training dynamics of GRPO and RLRT from the base policy and from the RSD-trained policy.** Average pass@1 on AIME 2024–2026 over the course of training for (a) Qwen3-4B and (b) Qwen3-8B, under the same training configuration. The two runs start from the same point, since RSD uses no ground truth and leaves pass@1 unchanged (Figure 2-(a)), yet the RSD-initialized run learns faster and ends higher.

6.2 A Better Attempt Boosts the Loop: Composing RSD with RLVR and Self-Distilled RLVR

We now test whether the improved attempt makes the policy a better learner in downstream on-policy training. We therefore run two instances of the loop in Table 1 that differ in the signal that drives the update. **GRPO** (Shao et al., 2024), *an RLVR method*, uses the verifier's reward alone. **RLRT** (Kim et al., 2026), *a self-distilled RLVR method*, adds the token-level self-distillation signal of OPSD. The same model, with the reference solution in its prompt, scores the policy's own rollouts, and the resulting signal reweights the advantages of the correct ones (Appendix B.3). Both are run from the base policy p^{θ_0} and from $p^{\theta_{\text{RSD}}}$ with the same training configuration and number of optimizer steps. Note that RSD enters only through the initialization, since its objective is not used during on-policy learning. RSD+GRPO thus denotes GRPO started from $p^{\theta_{\text{RSD}}}$, and it differs from Base+GRPO only in the attempt distribution the loop starts from (likewise for RLRT).

Starting from the RSD-trained policy improves the final model under both schemes and at both scales (Table 2). Averaged over the five benchmarks, pass@1/pass@4 rise by $+1.6/+1.2$ (4B) and $+8.3/+8.4$ (8B) under GRPO and by $+8.6/+6.5$ (4B) and $+9.5/+8.0$ (8B) under RLRT.

Figure 3 shows the training dynamics of the GRPO and RLRT runs at both scales in detail. The two initializations start at the same accuracy, and the RSD-initialized run moves ahead within the first few checkpoints. Part of the explanation lies in the learning signal itself. A prompt yields a nonzero GRPO advantage only if at least one of its G rollouts succeeds, so the fraction of prompts that provide a signal is bounded by the initialization's pass@G (G=8 here). Since the two runs of each pair differ only in their initialization, these results support our central claim that retrospection makes a better learner. Appendices D.2 and D.3 extend the comparison to **OPD** from a larger external teacher and to **OPSD**.

6.3 FLOP-Matched Comparison

Although RSD requires no ground-truth answer, its warmup and distillation stages consume compute before reinforcement learning begins. The comparison in Section 6.2 ignores this cost, since it matches the two runs of each pair in RL steps rather than in total compute. We quantify it in FLOPs, counting each generated or teacher-scored token as one forward pass and each trained token as three (Appendix B.4). On this measure the two stages correspond to 45–56 additional GRPO steps at 4B and 61–83 at 8B.

We therefore continue the Base+GRPO runs to the FLOP-matched budget, to step 160 at 4B and step 180 at 8B, and compare them with RSD+GRPO at step 100 (Table 3). At 8B the extra steps raise the base run from 33.0 to 38.6 average pass@1 on AIME 2024–2026, but RSD+GRPO stays ahead by 4.9 points. At 4B the base run does not improve past step 100, falling from 27.8 to 26.7, and the margin widens from 1.9 to 3.0 points. The advantage of the RSD initialization thus remains when the base policy is given the compute that RSD itself consumed.

Table 3: **The RSD advantage persists under FLOP matching.** Pass@1 and pass@4 (%) on AIME 2024–2026 for Base+GRPO at step 100 (as in Table 2) and at the FLOP-matched budget (step 160 at 4B, step 180 at 8B), and for RSD+GRPO at step 100. Avg. is the mean over the three sets.

Model	Method	AIME24		AIME25		AIME26		Avg.	
		pass@1	pass@4	pass@1	pass@4	pass@1	pass@4	pass@1	pass@4
Qwen3-4B	Base + GRPO, step 100	31.1	50.6	24.0	39.1	28.3	43.9	27.8	44.5
	Base + GRPO, step 160 (FLOP-matched)	27.4	48.1	24.9	39.6	27.7	42.9	26.7	43.5
	RSD + GRPO, step 100 (Ours)	**32.0**	**51.3**	**26.9**	**40.1**	**30.1**	**44.5**	**29.7**	**45.3**
Qwen3-8B	Base + GRPO, step 100	39.4	62.8	30.4	44.4	29.3	50.2	33.0	52.5
	Base + GRPO, step 180 (FLOP-matched)	43.8	67.9	34.3	48.9	37.6	55.8	38.6	57.5
	RSD + GRPO, step 100 (Ours)	**49.2**	**71.4**	**39.2**	**57.8**	**42.1**	**61.4**	**43.5**	**63.5**

7 Related Work

On-Policy (Self-)Distillation. Knowledge distillation (Hinton et al., 2015; Kim & Rush, 2016), and in particular its on-policy form, which supervises the student on its own rollouts (Gu et al., 2024; Agarwal et al., 2024; Jin et al., 2026), is a standard stage of post-training (Yang et al., 2025; Yoon et al., 2026; Liu et al., 2026). OPSD removes the external teacher by conditioning the same model on privileged information (Zhao et al., 2026; Wang et al., 2026; Li et al., 2026b): a reference solution, environment feedback or peer rollouts (Hübotter et al., 2026; Yu et al., 2026b), a reflective trajectory anchored to the correct answer (Huang et al., 2026), or a critique from a stronger frozen model (Kara & Ersoy, 2026). All of these require labels, a verifier, or an external model, except two variants aimed at long-context modeling (Zhang et al., 2026) and reasoning compression (Sang et al., 2026). Two findings shape our design: on-policy distillation tends to shorten reasoning (Kim & Lee, 2026) and to *contract* large-k coverage even as average accuracy rises (Ge et al., 2026), and it works best when the teacher's context aligns with the student's own trace (Kara & Ersoy, 2026). RSD takes this alignment to its limit, since the teacher's context is the student's own prior attempt, and it addresses coverage at the attempt, where coverage is set.

Label-Free Self-Improvement. Self-consistency (Wang et al., 2023) has been turned into a label-free training signal that reinforces agreement among a model's own outputs (Zuo et al., 2026; Prasad et al., 2025; Shafayat et al., 2025) or between its behavior and its own explanations of it (Pres et al., 2026). A vote, however, cannot reinforce the correct minority rollouts on which hard problems are solved, and vote-based training is prone to collapse (Yu et al., 2025). Closest to RSD, Samanta et al. (2025) post-train a model on multi-agent debate, internalizing behavior conditioned on peers' attempts, but with supervision from vote-filtered preferences, and He et al. (2026) distill a self-conditioned policy into the unconditioned one, but their teacher also sees the binary reward of the response it revises. RSD distills a dense per-token distribution from the model's own attempt alone, with no vote and no reward, and expands the set of solutions that bootstrapping (Zelikman et al., 2022) and RLVR (Yue et al., 2025) can then exploit. Gandhi et al. (2025) find that behaviors present in the initial policy, such as verification and backtracking, govern how much it improves under reinforcement learning, and induce them from a stronger model's traces or curated pretraining data; RSD likewise targets the initial policy, but with the model's own attempts alone.

8 Conclusion

The post-training of reasoning models follows an attempt–observe–update loop, and we target its first stage. Because attempting precedes observing, the attempt can be improved only from the model itself, and we show that retrospection expands coverage while preserving single-attempt accuracy. Direct distillation of the conditioned policy fails because its reasoning collapses in length, and RSD repairs this with a brief teacher warmup before transferring the behavior into a single generation pass. The resulting policy is a better learner, in that RLVR (GRPO) and self-distilled RLVR (RLRT) started from it reach higher pass@1 and pass@4 across five benchmarks and two model scales, and the GRPO advantage on AIME persists after matching FLOPs. These results indicate that the attempt stage is trainable and that a loop built on a better attempt learns more from the subsequent observe and update stages.

AI Use Statement

In this work, we used generative AI tools to polish the writing, to identify relevant literature, to assist with the figures, and to assist with the implementation of the method and the experiment infrastructure. We have not used generative AI tools for research ideation, experimental design, or the analysis of results, and the other tasks on the required-disclosure list are not applicable to this work. We have reviewed all AI-assisted work, and all AI-assisted code was verified and tested for correctness by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

Ethics Statement

This work studies post-training methods for improving mathematical reasoning in publicly available language models and uses only public benchmark datasets; it involves no human subjects, personal data, or sensitive applications. Improved reasoning capability is broadly dual-use, but we do not believe RSD introduces risks beyond those of the underlying models and standard post-training practice.

Reproducibility Statement

We describe the complete training procedure in Section 4 and Algorithm 1, with all hyperparameters, prompt templates, and infrastructure details in Appendices B and C. Evaluation follows the unbiased pass@k estimator of Chen et al. (2021) with sampling configurations specified in Section 5.

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*, volume 2024, pp. 21246–21263, 2024.

Mislav Balunovic, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. Matharena: Evaluating llms on uncontaminated math competitions. *Advances in Neural Information Processing Systems*, 38, 2026.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D. Goodman. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars, 2025. URL https://arxiv.org/abs/2503.01307.

Xinmu Ge, Zizhuo Zhang, Yu Huang, Jianing Zhu, Lin Yuan, Wanli Gu, Weichang Wu, Weiran Huang, Xiaolu Zhang, Bo Han, Jun Zhou, and Jiangchao Yao. Towards understanding on-policy distillation through the lens of test-time scaling. *arXiv preprint arXiv:2608.11829*, 2026.

H. P. Grice. Logic and conversation. In Peter Cole and Jerry L. Morgan (eds.), *Syntax and Semantics: Vol. 3: Speech Acts*, pp. 41–58. Academic Press, New York, 1975. URL http://www.ucl.ac.uk/ls/studypacks/Grice-Logic.pdf.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *International Conference on Learning Representations*, volume 2024, pp. 32694–32717, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Yinghui He, Simran Kaur, Adithya Bhaskar, Yongjin Yang, Jiarui Liu, Narutatsu Ri, Liam Fowl, Abhishek Panigrahi, Danqi Chen, and Sanjeev Arora. Self-distillation zero: Self-revision turns binary rewards into dense supervision. *arXiv preprint arXiv:2604.12002*, 2026.

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

Zhilin Huang, Hang Gao, Ziqiang Dong, Yuan Chen, Yifeng Luo, Chujun Qin, Jingyi Wang, Yang Yang, and Guanjun Jiang. Learning from your own mistakes: Constructing learnable micro-reflective trajectories for self-distillation. *arXiv preprint arXiv:2606.18844*, 2026.

Jonas Hübotter, Frederike Lübeck, Lejs Deen Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. Reinforcement learning via self-distillation. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=QkfkxyRizZ.

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.

Woogyeol Jin, Taywon Min, Yongjin Yang, Dennis Wei, Yi Zhou, Swanand Ravindra Kadhe, Nathalie Baracaldo, and Kimin Lee. Entropy-aware on-policy distillation of language models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=J5i09faOOf.

Manu Kapur. Productive failure in learning math. *Cognitive science*, 38(5):1008–1022, 2014.

Semih Kara and Oğuzhan Ersoy. The role of feedback alignment in self-distillation. *arXiv preprint arXiv:2606.11173*, 2026.

Jaehoon Kim and Dongha Lee. Opsd compresses what rlvr teaches: A post-rl compaction stage for reasoning models. *arXiv preprint arXiv:2605.06188*, 2026.

Jeonghye Kim, Jiwon Jeon, Dongsheng Li, and Yuqing Yang. Rebellious student: Reversing teacher signals for reasoning exploration with self-distilled rlvr. *arXiv preprint arXiv:2605.10781*, 2026.

Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 1317–1327, 2016.

Aitor Lewkowycz, Anders Johan Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Venkatesh 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 Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=IFXTZERXdM7.

Gengsheng Li, Tianyu Yang, Junfeng Fang, Mingyang Song, Mao Zheng, Haiyun Guo, Dan Zhang, Jinqiao Wang, and Tat-Seng Chua. Unifying group-relative and self-distillation policy optimization via sample routing. In *Third Conference on Language Modeling*, 2026a. URL https://openreview.net/forum?id=P2OuWwZspP.

Pengyu Li, Zhitao Gao, Lingling Zhang, Muye Huang, Yuanming Li, Fangzhi Xu, and Jun Liu. Visual-opsd: Cross-modal on-policy self-distillation for efficient unified multimodal reasoning. *arXiv preprint arXiv:2606.18974*, 2026b.

Ruiqi Liu, Xiaolei Lv, Gengsheng Li, Ximo Zhu, Zhiheng Wang, Zhengbo Zhang, Junkai Chen, Zhiheng Li, Bo Li, Jun Gao, et al. Visual-advantage on-policy distillation for vision-language models. *arXiv preprint arXiv:2605.21924*, 2026.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=5PAF7PAY2Y.

Roy Lyster. Differential effects of prompts and recasts in form-focused instruction. *Studies in Second Language Acquisition*, 26(3):399–432, 2004. doi: 10.1017/S0272263104263021.

Archiki Prasad, Weizhe Yuan, Richard Yuanzhe Pang, Jing Xu, Maryam Fazel-Zarandi, Mohit Bansal, Sainbayar Sukhbaatar, Jason E Weston, and Jane Yu. Self-consistency preference optimization. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=94G4eL3RWi.

Itamar Pres, Laura Ruis, Melat Ghebreselassie, Belinda Z Li, and Jacob Andreas. Self-ctrl: Self-consistency training with reinforcement learning. *arXiv preprint arXiv:2606.18327*, 2026.

Ankur Samanta, Akshayaa Magesh, Runzhe Wu, Ayush Jain, Youliang Yu, Daniel Jiang, Boris Vidolov, Paul Sajda, Yonathan Efroni, and Kaveh Hassani. Self-improvement of language models by post-training on multi-agent debate. *arXiv preprint arXiv:2509.15172*, 2025.

Hejian Sang, Yuanda Xu, Zhengze Zhou, Ran He, Zhipeng Wang, and Jiachen Sun. Crisp: Compressed reasoning via iterative self-policy distillation. *arXiv preprint arXiv:2603.05433*, 2026.

Sheikh Shafayat, Fahim Tajwar, Ruslan Salakhutdinov, Jeff Schneider, and Andrea Zanette. Can large reasoning models self-train? *arXiv preprint arXiv:2505.21444*, 2025.

Yingyu Shan, Yuhang Guo, Zihao Cheng, Zeming Liu, Xiangrong Zhu, Xinyi Wang, Jiashu Yao, Wei Lin, Hongru Wang, and Heyan Huang. Learning from own solutions: Self-conditioned credit assignment for reinforcement learning with verifiable rewards. *arXiv preprint arXiv:2606.18810*, 2026.

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

Sihan Wang, Xiyao Liu, Lianqing Liu, and Zhi Han. Seeing before reasoning: Decoupling perception and reasoning for shortcut-resilient multimodal on-policy self-distillation. *arXiv preprint arXiv:2606.19120*, 2026.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

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.

Eunseop Yoon, Hee Suk Yoon, Jaehyun Jang, SooHwan Eom, Qi Dai, Chong Luo, Mark A Hasegawa-Johnson, and Chang D Yoo. Pacr: Progressively ascending confidence reward for llm reasoning. *arXiv preprint arXiv:2510.22255*, 2025.

Hee Suk Yoon, Eunseop Yoon, Jaehyun Jang, SooHwan Eom, Ji Woo Hong, Mark A. Hasegawa-Johnson, Qi Dai, Chong Luo, and Chang D. Yoo. Decomposed on-policy distillation for vision-language reasoning: Steering gradients for visual grounding. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=Jb1YBPqyb7.

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

Weichen Yu, Xiaomin Li, Yizhou Zhao, Xiaoze Liu, Ruowang Zhang, Haixin Wang, Yinyi Luo, Chen Henry Wu, Gaurav Mittal, Matt Fredrikson, et al. Multi-rollout on-policy distillation via peer successes and failures. *arXiv preprint arXiv:2605.12652*, 2026b.

Zhaoning Yu, Will Su, Leitian Tao, Haozhu Wang, Aashu Singh, Hanchao Yu, Jianyu Wang, Hongyang Gao, Weizhe Yuan, Jason Weston, et al. Restrain: From spurious votes to signals–self-driven rl with self-penalization. *arXiv preprint arXiv:2510.02172*, 2025.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=4OsgYD7em5.

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

Xinsen Zhang, Zhenkai Ding, Tianjun Pan, Run Yang, Chun Kang, Xue Xiong, and Jingnan Gu. Opsdl: On-policy self-distillation for long-context language models. *arXiv preprint arXiv:2604.17535*, 2026.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=Jpxfof0EaS.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Xinwei Long, Ermo Hua, Biqing Qi, et al. Ttrl: Test-time reinforcement learning. *Advances in Neural Information Processing Systems*, 38:131459–131483, 2026.

A Limitations

This paper shows the effectiveness of retrospection in the language reasoning domain. Retrospection itself is not tied to a modality, since the teacher is conditioned only on the model's own prior attempt. An extension of RSD to vision-language reasoning, which would establish whether the same benefit holds there, is left to future work. The warmup and distillation stages of RSD consume compute before reinforcement learning begins. Section 6.3 shows that the advantage of the RSD initialization outweighs this cost, since it remains when the base run is given the same FLOPs, but the cost is one that must be budgeted for. Finally, the benefit of the RSD initialization is tied to the source of the learning signal. In GRPO and RLRT the signal comes from the model itself, as the verifier's reward on its attempts or its own solution-conditioned distribution, whereas under on-policy distillation from a larger external teacher it comes from that teacher, and there the initialization has little influence (Appendix D.2).

B Implementation Details

B.1 Software and Hardware

All training uses a fork of TRL that adds the on-policy distillation and RLRT trainers, with vLLM for rollout generation. Runs use DeepSpeed ZeRO-3 with optimizer states and parameters offloaded to CPU, in bfloat16 with FlashAttention-2 and gradient checkpointing. The fork and the exact library versions will be released with the code.

Rollouts are generated by a colocated vLLM engine (sleep mode; memory fraction 0.7 for distillation and 0.8 for reinforcement learning) at temperature 1.0 with top-p 1.0 and no top-k, in non-thinking mode, with prompts capped at 2,048 and completions at 8,192 tokens. The optimizer is fused AdamW (β_1=0.9, β_2=0.999, ϵ=10^{-8}, no weight decay) with gradient-norm clipping at 1.0 and a constant learning rate without warmup. The seed is 42 throughout. All runs use a single node with 8× NVIDIA A100 GPUs.

B.2 Distillation Stages

Table 4 lists the hyperparameters of the teacher warmup and the retrospective distillation stages.

B.3 Reinforcement-Learning Stages

GRPO. We use an answer-correctness reward only (no format reward), group-normalized advantages, and token-level loss normalization as in DAPO (Yu et al., 2026a), with the symmetric clip

Table 4: Hyperparameters of the two distillation stages.

	Teacher warmup		RSD distillation	
Hyperparameter	Qwen3-4B	Qwen3-8B	Qwen3-4B	Qwen3-8B
Initialization	Qwen3-4B	Qwen3-8B	Qwen3-4B	Qwen3-8B
Teacher	initial policy	initial policy	warmup step 20	warmup step 20
Objective	generalized JSD, β=0.5, temperature 1.0			
Learning rate	1×10^{-6}, constant, no warmup			
Prompts per optimizer step	128	128	128	128
Batch × accumulation × GPUs	$1 \times 16 \times 8$	$1 \times 16 \times 8$	$1 \times 16 \times 8$	$1 \times 16 \times 8$
Max prompt / completion length	2,048 / 8,192 tokens			
Teacher prompt limit	32,256 tokens			
Rollout sampling	T=1.0, top-p 1.0, no top-k			
Checkpoint used	step 20	step 20	step 320	step 320

range $[0.8, 1.2]$ and without dynamic sampling or overlong filtering. We do not use the KL penalty term, so no reference model is used. An importance-sampling correction between the generation and training policies is capped at 3.0 with sequence-level masking.

RLRT. RLRT (Kim et al., 2026) shares the GRPO configuration and adds a per-token reweighting of correct rollouts. With P_S the policy and P_T the same policy evaluated with a verified solution in its context, the advantage of token t is multiplied by

$$m_t = (1 - \lambda) + \lambda \,\mathrm{clip}\big(P_S(y_t)/P_T(y_t),\, 1 - \epsilon_w,\, 1 + \epsilon_w\big),$$

with λ=0.5 and ϵ_w=0.5, so that tokens on which the policy departs from the solution-conditioned teacher receive larger weight, the reversed teacher signal of Kim et al. (2026); no gradient flows through the teacher pass. The verified solutions come from a pool of 12,741 DAPO problems with one to four verified-correct Qwen3-32B traces each (2.89 on average), one trace resampled at each encounter, under a teacher prompt limit of 12,288 tokens.

Runs. All eight runs (two schemes, two scales, two initializations) use the hyperparameters of Table 5 and are evaluated at step 100.

Table 5: Hyperparameters of the reinforcement-learning stages. Both schemes share the backbone, data, and reward; RLRT adds the privileged-context reweighting.

Hyperparameter	GRPO	RLRT
Learning rate	1×10^{-5}, constant	1×10^{-5}, constant
Rollouts per prompt	8	8
Prompts per optimizer step	128 (1,024 completions)	128 (1,024 completions)
Generation batch	4,096 completions	4,096 completions
Loss	token-level (DAPO-style)	token-level (DAPO-style)
Clip range	$[0.8, 1.2]$, symmetric	$[0.8, 1.2]$, symmetric
KL coefficient	0 (no reference model)	0 (no reference model)
Reward	answer correctness only	answer correctness only
Importance-sampling cap	3.0, sequence-level mask	3.0, sequence-level mask
Max prompt / completion length	2,048 / 8,192 tokens	2,048 / 8,192 tokens
Privileged context	—	one verified trace, 12,288-token limit
Weight clip ϵ_w	—	0.5
Mixing coefficient λ	—	0.5, constant
Checkpoint used	100	100

B.4 Compute Accounting

The cost of the warmup and distillation stages is expressed in parameter-tokens: each token generated by the rollout engine or scored by a teacher forward pass is counted as $2P$ FLOPs and each

token trained on as $6P$, with P the parameter count, so that one parameter-token corresponds to one forward pass over one token. Counting the prior-attempt and learning rollouts of each prompt, the teacher's forward pass, and the student's forward and backward passes, the two stages cost 1,345M parameter-tokens for Qwen3-4B and 2,196M for Qwen3-8B. Measured the same way against the logged cost of steps 101–200 of the GRPO runs, the two stages correspond to 45–56 additional GRPO steps at 4B and 61–83 at 8B, the ranges reflecting the variation of the per-step cost over the run; matching by logged GPU-hours instead gives 95–101 and 83–90 additional steps. Table 3 uses step 160 at 4B, beyond the FLOP-matched range of 145–156 steps, and step 200 at 8B, beyond both the FLOP-matched range of 161–183 steps and the GPU-hour-matched range of 183–190 steps. Counting generated tokens alone would overstate the cost of RSD, since the prior-attempt rollouts are decoded without a gradient, and counting trained tokens alone would understate it, since it ignores the teacher forward passes.

B.5 Evaluation Protocol

Benchmarks. We use AIME 2024, AIME 2025, AIME 2026 and HMMT February 2026 (Balunovic et al., 2026), and Minerva-Math (Lewkowycz et al., 2022).

Decoding and Sample Counts. Decoding uses temperature 0.7, top-p 0.8, and top-k 20, the recommended setting for Qwen3 in non-thinking mode, with no fixed sampling seed. Samples are drawn in independent blocks of 256; four blocks are pooled to $N{=}1{,}024$ for the analysis of Section 3 and the RSD-alone results (a duplicate check found no identical generations among 23,040 teacher samples). Reinforcement-learning checkpoints are evaluated with $N{=}64$; for the GRPO runs on AIME and HMMT, these are the first 64 of 256 generated samples.

Maximum Generation Length. Base and RSD policies are decoded with a maximum generation length of 16,384 tokens; the GRPO and RLRT runs with 32,768; and the conditioned policy with 6,144, its context additionally holding the prior attempt, truncated at evaluation, within a total context of at most 40,960 tokens. Comparisons are made only between arms evaluated with the same maximum generation length.

Estimator. Pass@k is estimated as $1 - \binom{n-c}{k} / \binom{n}{k}$ per problem and averaged over problems (Chen et al., 2021).

Grading. The answer is the last `\boxed{}` in the completion, compared to the reference with the robust grader of Liu et al. (2025), which accepts equivalent surface forms (e.g., `042`, `42.0`, `x=42`) and counts an unboxed answer as wrong.

C Prompt Templates

Table 6 gives the prompts verbatim. All of them are rendered by the Qwen3 chat template with `enable_thinking=False`, which inserts an empty `<think></think>` block at the start of the assistant turn, so that generation begins after it; the reasoning the prompts ask for is the `<reason>` block. This prompt is used unchanged for the base model, for the student in both distillation stages, for the GRPO and RLRT policies, and for every evaluation without a prior attempt in context. The teacher prompt of RSD places the prior attempt in the user turn in full. RLRT and the OPSD ablation in Appendix D.3 use the same structure with a verified solution in place of the prior attempt.

Table 6: Prompt templates. Placeholders in braces are filled at run time.

Base model, student, and evaluation prompt (system turn; the user turn is the problem statement)

```
A conversation between user and assistant. The user asks a question,
and the assistant solves it. The assistant first thinks about the
reasoning process in the mind and then provides the user with the
answer. The reasoning process should be enclosed within <reason>
</reason> tags. The final answer MUST BE put in \boxed{}.
```

RSD teacher prompt (system turn)

```
A conversation between user and assistant. The user asks a question
and shares several sample responses from previous attempts. The
assistant uses these samples as informational context to understand
typical reasoning patterns and approaches to the question, then
produces its own coherent response that integrates the patterns
demonstrated. The assistant does not simply copy any single sample
but instead synthesizes a response informed by all of them. The
reasoning process should be enclosed within <reason> </reason> tags.
The final answer MUST BE put in \boxed{}.
```

RSD teacher prompt (user turn)

```
Question: {question}
Reference samples from previous attempts:
Sample 1:
{prior attempt 1}
```

RLRT teacher prompt (system turn)

```
A conversation between user and assistant. The user asks a question
and shares several reference solutions to the question. The assistant
uses these solutions as informational context to understand effective
reasoning patterns and approaches to the question, then produces its
own coherent response that integrates the patterns demonstrated.
The assistant does not simply copy any single solution but instead
synthesizes a response informed by all of them. The reasoning process
should be enclosed within <reason> </reason> tags. The final answer
MUST BE put in \boxed{}.
```

RLRT teacher prompt (user turn)

```
Question: {question}
Reference solutions:
Solution 1:
{verified solution}
```

OPSD teacher prompt (for Appendix D.3)

Identical to the RLRT teacher prompt above.

D Additional Results

D.1 Analysis for Qwen3-4B

Figure 4 repeats the analysis of Figure 2 for Qwen3-4B, and the findings are the same as for Qwen3-8B. Conditioning shortens the reasoning from 3.5k to 0.9k tokens, the *naive* RSD student inherits the collapsed form (1.6k) and falls below the base policy, and RSD matches the conditioned policy while reasoning at full length (3.8k).

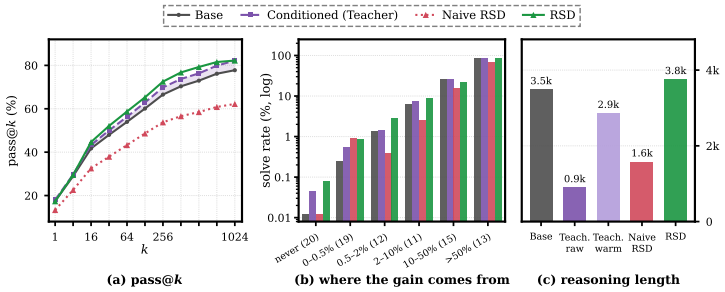

Figure 4: **Analysis of Figure 2 for Qwen3-4B.** The findings are the same as for Qwen3-8B. **(a)** Retrospection expands coverage at large k, and RSD transfers it into a single generation pass whereas the *naive* student falls below the base policy. **(b)** The gain comes from the problems the base policy rarely solves. **(c)** Conditioning collapses the reasoning length, the naive student inherits the collapse, and RSD reasons at full length.

D.2 Initialization Under OPD from a Larger External Teacher

The loops of Section 6.2 involve no external teacher, since GRPO learns from the verifier's reward and RLRT from the model's own solution-conditioned distribution. OPD (Table 1) likewise trains on the student's own rollouts, but its signal comes from a larger model, the *external teacher*. Its next-token distribution supervises every position of the rollout whether or not the student's attempt succeeds, so the student is pulled toward it from wherever it starts, and one would expect the initialization to matter less than in the loops of Section 6.2.

We test this with the Qwen3-8B Base+GRPO policy of Table 2 as the external teacher, distilled into Qwen3-4B with Eq. 2 under the reverse KL divergence ($\beta \to 1$ in Eq. 1), once from the base policy and once from $p^{\theta_{\mathrm{RSD}}}$ under the same configuration (Table 7). The two students perform similarly, with pass@1 and pass@4 on AIME 2024–2026 within 0.1 points of each other, and both come close to the external teacher. The initialization thus has little influence under an external teacher, and matters most where the model is its own source of supervision and the attempt distribution bounds the learning signal, the setting in which RSD improves the loop in Section 6.2.

Table 7: **The initialization has little influence under on-policy distillation from a larger external teacher.** Pass@1 and pass@4 (%) on AIME 2024–2026 of Qwen3-4B students distilled by OPD (Eq. 2) from the Qwen3-8B Base+GRPO policy of Table 2 as the *external teacher*, starting from the base policy or from $p^{\theta_{\mathrm{RSD}}}$.

Policy	AIME24		AIME25		AIME26		Avg.	
	pass@1	pass@4	pass@1	pass@4	pass@1	pass@4	pass@1	pass@4
Qwen3-8B Base + GRPO (external teacher)	39.4	62.8	30.4	44.4	29.3	50.2	33.0	52.5
Qwen3-4B								
On-policy distillation (OPD) (Agarwal et al., 2024)								
Base + OPD	34.4	54.6	26.5	40.5	27.2	45.3	29.4	46.8
RSD + OPD	34.2	54.8	26.8	40.5	27.5	44.7	29.5	46.7
Δ	−0.2	+0.2	+0.3	0.0	+0.3	−0.6	+0.1	−0.1

Table 8: **Initialization from the same two-stage procedure with a reference solution in place of the prior attempt.** Pass@1 and pass@4 (%) of Qwen3-4B after GRPO and after RLRT, started from the base policy, from the OPSD checkpoint trained on the reference solution τ^*, and from the RSD-trained policy, under the configuration of Table 2.

Method	AIME24		AIME25		AIME26		HMMT26		Minerva		Avg.	
	pass@1	pass@4	pass@1	pass@4	pass@1	pass@4	pass@1	pass@4	pass@1	pass@4	pass@1	pass@4
Qwen3-4B												
RLVR (GRPO) (Shao et al., 2024)												
Base + GRPO	31.1	50.6	24.0	39.1	28.3	43.9	18.6	28.9	61.2	67.9	32.6	46.1
OPSD + GRPO	28.7	40.4	22.2	36.6	20.9	37.2	20.3	29.7	**61.6**	69.6	30.7	42.7
RSD + GRPO (Ours)	**32.0**	**51.3**	**26.9**	**40.1**	**30.1**	**44.5**	**21.1**	**30.8**	60.9	**69.8**	**34.2**	**47.3**
Self-distilled RLVR (RLRT) (Kim et al., 2026)												
Base + RLRT	47.0	65.9	40.8	63.5	37.9	57.5	29.1	39.1	62.0	70.8	43.4	59.4
OPSD + RLRT	52.5	67.4	42.8	59.4	45.2	60.4	29.7	37.6	62.6	71.3	46.6	59.2
RSD + RLRT (Ours)	**59.1**	**74.6**	**51.7**	**67.9**	**51.4**	**71.8**	**33.2**	**41.6**	**64.7**	**73.5**	**52.0**	**65.9**

D.3 A Reference-Solution Control: The Same Procedure with OPSD

Although Table 1 and Section 2.2 treat OPSD as an on-policy learning framework, the downstream experiments of Section 6.2 use GRPO and RLRT rather than OPSD itself, because training with the OPSD objective alone has been reported to be unstable for mathematical reasoning. On Qwen3-8B-Base, Kim et al. (2026) observe that the reward collapses to zero within roughly 20 steps and the response length diverges, an effect they attribute to the suppression of hedging and reflective tokens, and they accordingly omit OPSD and its variants (Zhao et al., 2026; Hübotter et al., 2026; Li et al., 2026a) from their base-model comparison. RLRT retains the OPSD signal but uses it only to reweight the GRPO advantages of correct rollouts, and this is the form in which the signal enters our experiments. In this section, OPSD serves solely as an initialization, trained under the same short schedule as RSD.

RSD differs from OPSD in what the teacher is shown: the model's own prior attempt τ^1 rather than a reference solution τ^*. To check that the benefit reported in Section 6.2 comes from retrospection rather than from the two-stage procedure itself, we rerun the procedure with the gold reference solution in place of the prior attempt. The warmup and distillation stages, their hyperparameters, and the step counts are those of RSD (Appendix B); the only changes are that τ^* replaces τ^1 in the teacher's context and that the teacher prompt states that the context is a reference solution (Appendix C).

We then initialize GRPO and RLRT for Qwen3-4B from this OPSD checkpoint, and compare the outcome with the base and RSD initializations. Table 8 reports the results and Figure 5 the training dynamics. Under GRPO, the OPSD initialization falls below the base initialization on average, whereas RSD improves on it. Under RLRT, whose learning signal is the same solution-conditioned distribution that OPSD distills, the OPSD initialization does help, but it remains well below the RSD initialization in both pass@1 and pass@4. *Conditioning the teacher on the model's own prior attempt is therefore more effective than conditioning it on a reference solution, which supports the use of* RSD *as an initialization for downstream on-policy learning.*

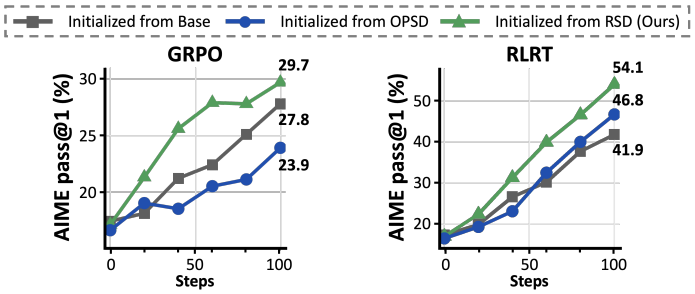

Figure 5: **Training dynamics of GRPO and RLRT from the base policy, the OPSD checkpoint, and the RSD-trained policy.** Average pass@1 on AIME 2024–2026 over the course of training for Qwen3-4B. Initializing from the OPSD checkpoint helps little, and hurts under GRPO, whereas initializing from RSD makes the policy learn faster and end higher under both algorithms.