How Should Teachers Be Prepared? RL on Student-Induced States for On-Policy Distillation

Anonymous authors
Paper under double-blind review

Abstract

On-policy distillation (OPD) improves the reasoning capabilities of small language models through token-level teacher supervision on student-generated trajectories. Yet can teachers that excel at solving problems independently also guide student reasoning effectively? Prior work shows that when student prefixes follow reasoning paths that differ from the teacher's own or contain errors, teachers can be less accurate when continuing from these prefixes than when solving problems independently. To this end, we propose Prep-OPD, which uses reinforcement learning (RL) before distillation to train the teacher to adapt to the student's existing reasoning state and correct course when errors arise. Training optimizes teacher continuations from fixed student prefixes using final-answer correctness as the reward. The prepared teacher then trains the student through trajectory guidance and token-level supervision. We evaluate Prep-OPD on eight mathematical reasoning benchmarks, using Qwen3-4B-Instruct-2507 as the teacher and Qwen3-0.6B and Qwen3-1.7B as students. With the 4B teacher and 1.7B student, Prep-OPD improves average accuracy over standard OPD and the strongest baseline, Relay-OPD, by 8.28 and 2.30 percentage points, respectively. Controlled experiments further show that teacher RL conditioned on student-generated prefixes yields higher student accuracy than problem-start teacher RL with and without handoff on Qwen3-1.7B. Reusing the same prepared teacher also improves Qwen3-0.6B.

1 Introduction

On-policy distillation (OPD) is a promising approach for transferring reasoning capabilities from strong language models to smaller students. The student generates its own trajectories, while the teacher provides dense, token-level supervision on student-generated prefixes, often through reverse KL minimization (Gu et al., 2024; Agarwal et al., 2024; Zhou et al., 2024). Training on student-generated trajectories helps reduce the distribution mismatch between the prefixes seen during training and those produced at inference (Lin et al., 2020). OPD can also achieve competitive reasoning performance at substantially lower training cost than a reinforcement learning (RL) baseline (Zhao et al., 2026; Jin et al., 2026).

However, OPD implicitly relies on the teacher being a reliable supervisor on student-generated reasoning states. This assumption can break down when student prefixes follow reasoning paths different from those the teacher would take independently, or have already gone in the wrong direction. In such cases, the teacher can be substantially less accurate when continuing from these prefixes than when solving the same problems from scratch. Its token-level feedback may also be noisy or even misleading (Koo et al., 2025; Fu et al., 2026; Xie et al., 2026b). Thus, strong independent problem-solving ability does not ensure reliable guidance on student prefixes (Figure 1(a)).

Existing work improves teacher guidance in several ways. Veto stabilizes optimization through adaptive distillation targets, but remains constrained by the teacher's existing continuation capabilities (Jang et al., 2026). ESR and FastOPD concentrate supervision on early prefixes where teacher guidance is more effective, but exclude reasoning steps beyond the truncation point from distillation supervision for that training instance (Zhou et al., 2026; Zhang et al., 2026). TRD improves distil-

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.

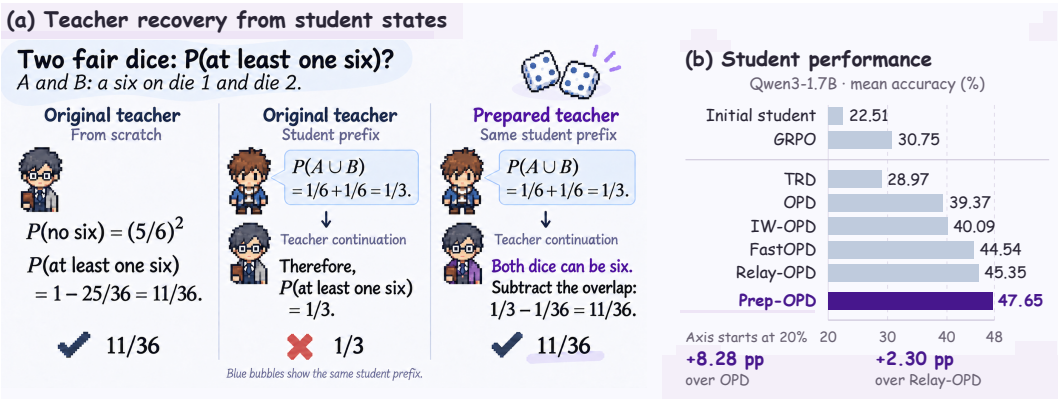


Figure 1: **Motivation and main results.** (a) A solution from scratch and two continuations from the same erroneous student prefix. (b) Qwen3-1.7B mean benchmark accuracy (%; Table 1).

lation samples by rewriting complete student trajectories, whereas SKD and Relay-OPD introduce teacher-generated tokens or segments during rollouts to intervene in student trajectories (Jiang et al., 2026; Xu et al., 2025; 2026). However, even with early intervention, teacher continuations may still be influenced by errors in the student's reasoning. These methods do not explicitly train the teacher to adapt to the student's existing reasoning state and correct course when errors arise.

This suggests a simple but different strategy: **prepare the teacher for student-generated reasoning states before using it for distillation**. To this end, we propose Prep-OPD, which uses reinforcement learning to train the teacher to adapt to the student's existing reasoning state and correct course when errors arise. We then distill this ability into the student. We sample prefixes at different reasoning stages from trajectories generated by the initial student. The teacher continues reasoning while preserving each prefix and is trained through reinforcement learning with final-answer correctness as the reward, yielding the prepared teacher T_P. We then freeze and reuse the same T_P: it supplies bounded continuations at selected intermediate states in fresh student rollouts and provides token-level distillation supervision over the entire trajectory jointly generated by the student and teacher. The teacher's continuations both provide context for the student's subsequent reasoning and contribute directly to its updates, allowing the student to learn to build on existing reasoning and adjust its direction when needed. Inference uses only the student model (Figure 2).

- Through controlled continuation experiments, we find that **teachers can be substantially less accurate when continuing from student prefixes than when solving independently, and this gap persists even at early prefix positions.** Problem-start RL improves the teacher's independent problem-solving accuracy but provides limited gains in continuation from student prefixes. Compared with the problem-start RL teacher, the prepared teacher narrows this gap and improves mean continuation accuracy by **2.55 percentage points**.
- We propose **Prep-OPD**, which uses reinforcement learning to prepare the teacher before distillation. Conditioned on student-generated reasoning prefixes and rewarded for final-answer correctness, the teacher learns to adapt to the student's existing reasoning state and correct course when errors arise. The prepared teacher then distills its learned continuation ability into the student through trajectory guidance and token-level supervision.
- Under a unified evaluation protocol, we evaluate Prep-OPD on eight mathematical reasoning benchmarks with two student sizes. With the 4B teacher and 1.7B student, Prep-OPD improves average accuracy over standard OPD and the strongest baseline, Relay-OPD, by **8.28 and 2.30 percentage points**, respectively.

2 Teacher Supervision on Student Trajectories

On-policy distillation (OPD) trains a student on trajectories sampled from its own policy, while a fixed teacher provides token-level supervision (Gu et al., 2024; Agarwal et al., 2024; Xu et al., 2026).

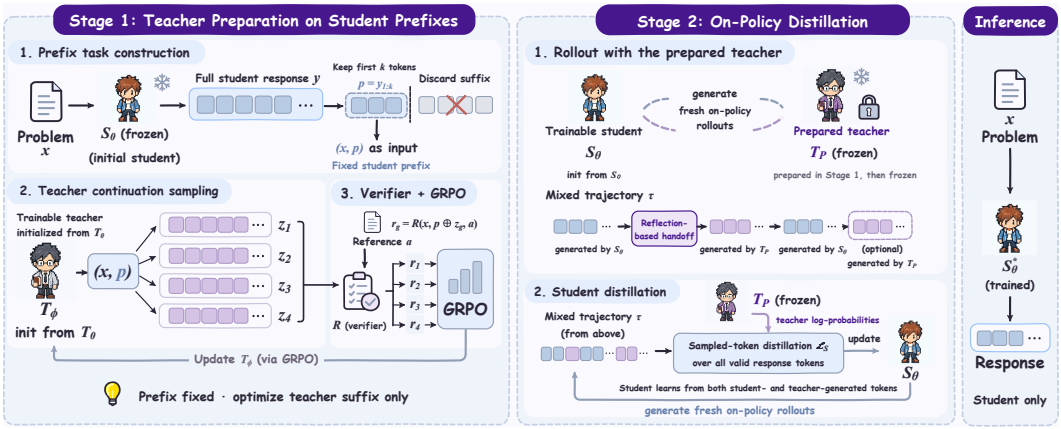


Figure 2: **Overview of Prep-OPD.** Stage 1 fixes prefixes from S_0 and trains T_ϕ, initialized from T_0, with outcome rewards and suffix-only GRPO. Stage 2 freezes T_P for bounded handoffs and token-level supervision of mixed trajectories. Blue/purple tokens denote student/teacher generation; distillation covers both token types. Inference uses the student alone.

Let S_θ denote the student, $S_{\bar{\theta}}$ its rollout policy, and T the teacher. Given a problem x, the student generates

$$y = (y_1, \ldots, y_N) \sim S_{\bar{\theta}}(\cdot \mid x), \qquad h_t = (x, y_{<t}). \tag{1}$$

Standard OPD typically employs the reverse KL divergence. Taking a single-sample estimate at the student-sampled token y_t, the teacher–student log-probability difference defines the token-level advantage

$$\widehat{A}_t^{\mathrm{OPD}} = \log T(y_t \mid h_t) - \log S_{\bar{\theta}}(y_t \mid h_t). \tag{2}$$

A positive advantage increases the probability of the sampled token, whereas a negative one decreases it. By conditioning supervision on h_t, OPD grounds learning in the reasoning states actually visited by the student rather than exclusively in teacher-generated trajectories.

However, student-generated reasoning may contain errors or commit to unproductive directions, which can compromise subsequent teacher supervision (Fu et al., 2026; Xie et al., 2026b). In such cases, the teacher must provide useful guidance while inheriting the reasoning already produced by the student, rather than solving the problem again from a clean starting point. A teacher's ability to solve problems from scratch therefore does not, by itself, establish its ability to provide effective supervision under student-generated reasoning. This motivates explicitly preparing the teacher to continue toward correct solutions from student-induced reasoning states.

3 PREP-OPD

To address this requirement, we propose **Prep-OPD**, a two-stage framework that first uses reinforcement learning to train the teacher to adapt to the student's existing reasoning state and correct course when errors arise, and then reuses the same prepared teacher to guide and supervise student learning. In **Stage 1**, we condition the teacher on student-generated reasoning prefixes and optimize only its continuations, using final-answer correctness as the reward, yielding the prepared teacher T_P (Sections 3.1–3.2). In **Stage 2**, we freeze and reuse the same T_P to guide student trajectories through bounded continuations and provide token-level distillation supervision over the entire trajectory jointly generated by the student and teacher (Section 3.3). Figure 2 illustrates the overall framework.

3.1 STUDENT-CONDITIONED CONTINUATION TASKS

To prepare the teacher to adapt to reasoning states actually generated by the student, we extract unfinished reasoning prefixes from trajectories of a fixed initial student S_0 to construct student-conditioned continuation tasks. Let $\mathcal{D}$ denote a dataset of problems x paired with reference answers

a. For each problem, we first sample a student response and then retain an unfinished prefix corresponding to an intermediate reasoning state:

$$(x, a) \sim \mathcal{D}, \qquad y \sim S_0(\cdot \mid x), \qquad k \sim \kappa(\cdot \mid y), \qquad p = y_{1:k}, \tag{3}$$

where κ is a cut-point distribution over valid intermediate positions. Repeating this procedure yields a fixed empirical distribution q_0 over continuation tasks (x, p, a).

Each task asks the teacher to continue from the inherited student reasoning p, rather than solve the problem again from a clean starting point. The reference answer a is not exposed to the teacher and is used only to evaluate the resulting continuation. Importantly, p remains fixed throughout the task: the teacher cannot delete or rewrite reasoning already produced by the student, and any adjustment must therefore be expressed through subsequent tokens.

Sampling cut points at different depths exposes the teacher to student reasoning states from different stages of a trajectory, ranging from relatively early partial solutions to more developed reasoning contexts.

3.2 Teacher Preparation via Reinforcement Learning

Teacher preparation objective. Given continuation tasks sampled from q_0, we train the teacher to continue toward correct solutions while inheriting the student's existing reasoning p. Let T_0 denote the initial teacher, and initialize a trainable teacher T_ϕ from T_0. Rather than restarting from the original problem, the teacher continues directly from the inherited student state:

$$z \sim T_\phi(\cdot \mid x, p), \tag{4}$$

where z denotes the teacher-generated continuation. The completed response is $p \oplus z$, and we optimize

$$J_{\text{prep}}(\phi) = \mathbb{E}_{\substack{(x,p,a)\sim q_0 \\ z\sim T_\phi(\cdot|x,p)}} \left[R(x, p \oplus z, a) \right], \tag{5}$$

where $R \in \{0, 1\}$ evaluates final-answer correctness. This formulation separates what is inherited from what can still be changed: p fixes the reasoning already produced by the student, whereas z is the continuation controlled by the teacher. Final-answer correctness then rewards continuations that successfully carry the inherited student state toward a correct solution.

Suffix-only optimization. We optimize the above teacher preparation objective using GRPO (Shao et al., 2024). For each continuation task, the teacher snapshot $T_{\bar{\phi}}$ used to generate the current batch samples G continuations $\{z_g\}_{g=1}^G$ from the same student prefix. Their rewards and group-relative advantages are

$$r_g = R(x, p \oplus z_g, a), \qquad \widehat{A}_g = \frac{r_g - \mu_r}{\sigma_r + \eta}, \tag{6}$$

where μ_r and σ_r are the mean and standard deviation of $\{r_g\}_{g=1}^G$, and $\eta > 0$ is a numerical stabilizer. Since all continuations in the group share the same problem x and inherited reasoning p, $\widehat{A}_g$ compares alternative ways of continuing from the same student-generated state.

For token t in continuation z_g, we define

$$\rho_{g,t}^T(\phi) = \frac{T_\phi(z_{g,t} \mid x, p, z_{g,<t})}{T_{\bar{\phi}}(z_{g,t} \mid x, p, z_{g,<t})}, \tag{7}$$

and minimize the clipped surrogate objective

$$\mathcal{L}_T(\phi) = -\widehat{\mathbb{E}}_{\text{suffix}} \left[\min \left\{ \rho_{g,t}^T(\phi) \widehat{A}_g,\ \text{clip}\big(\rho_{g,t}^T(\phi), 1 - \epsilon_T, 1 + \epsilon_T\big)\, \widehat{A}_g \right\} \right]. \tag{8}$$

Here, $\widehat{\mathbb{E}}_{\text{suffix}}$ averages over teacher-generated suffix tokens, with x and p serving only as conditioning context. Rewards are evaluated on the complete response $p \oplus z_g$. We keep q_0 fixed throughout Stage 1 while sampling fresh teacher continuations, and denote the resulting frozen prepared teacher by T_P.

3.3 On-Policy Distillation with the Prepared Teacher

Having learned to continue from inherited student reasoning, the frozen T_P is reused in Stage 2 in two complementary roles: at selected states, it contributes continuations that steer the student rollout toward more reliable reasoning directions; over the resulting mixed trajectory, it provides token-level supervision.

At each handoff, the current trajectory prefix plays the same role as p in Stage 1: it is preserved as inherited reasoning, while T_P determines what comes next. Following Relay-OPD (Xu et al., 2026), we use bounded teacher handoffs to insert continuations at selected student prefixes. During a student segment, while the handoff budget remains, T_P takes over when its most likely next token is a reflection token (Muennighoff et al., 2025) that differs from the rollout student $S_{\bar{\theta}}$'s proposed next token at the same prefix. Equation 14 in Appendix A.2 specifies this criterion; the same appendix defines the reflection-token set and handoff budgets. At a handoff, T_P emits the preferred reflection token and then samples a bounded continuation, producing a mixed trajectory $\tau = (\tau_1, \ldots, \tau_N)$ containing both student- and teacher-generated tokens.

The same frozen T_P then provides OPD supervision over the mixed trajectory. For $h_t^\tau = (x, \tau_{<t})$, we define

$$\widehat{A}_t^P = \mathrm{sg}[\log T_P(\tau_t \mid h_t^\tau) - \log S_{\bar{\theta}}(\tau_t \mid h_t^\tau)], \tag{9}$$

where sg denotes stop-gradient. We optimize all realized response tokens with a clipped sampled-token distillation surrogate; the complete student update, token averaging, and rollout budgets are specified in Appendix A.

Teacher-generated continuations therefore both shape the contexts from which the student resumes generation and participate directly in the student update, while the same T_P provides dense token-level supervision over the full mixed trajectory. After distillation, T_P is removed and inference uses the student alone.

4 Experiments

4.1 Experimental Setup

Models and datasets. We use Qwen3-4B-Instruct-2507 as the initial teacher T_0 and Qwen3-1.7B as the primary student S_0, with an additional Qwen3-0.6B setting (Yang et al., 2025). Training problems come from the English subset of DAPO-Math-17K (Yu et al., 2025). Both student distillation settings use $16{,}164$ problems, while the GRPO stage of teacher preparation uses a $2{,}048$-problem training subset. We evaluate on the eight mathematical reasoning benchmarks used by Xu et al. (2026): AIME 2024, AIME 2025, AIME 2026, AMC 2023, HMMT February 2026, HMMT November 2025, OlympiadBench, and MATH-500.

Baselines. We include the initial student as a reference and compare with GRPO (Shao et al., 2024), Standard OPD (Gu et al., 2024; Agarwal et al., 2024), TRD (Jiang et al., 2026), IW-OPD (Xie et al., 2026b), FastOPD (Zhang et al., 2026), and Relay-OPD (Xu et al., 2026). In the main Qwen3-1.7B comparison, the distillation baselines use T_B, obtained by problem-start RL from T_0, whereas Prep-OPD uses T_P, trained on prefixes from S_0. The two RL-trained teachers use the same training-problem pool and outcome verifier. For Qwen3-0.6B, TRD uses T_0, while the other distillation baselines in Table 1 use T_B. Relay-OPD comparisons using T_0 at both student sizes are reported separately in Appendix A.6.

Implementation details. We implement training with verl and vLLM on H20 GPUs, with thinking disabled for both students. Teacher preparation runs for 128 GRPO steps; student distillation at both sizes uses one epoch. We use a $16{,}384$-token generation cap throughout training and evaluation; FastOPD truncates its training rollouts at $4{,}096$ tokens. Training and evaluation use temperature 1.0 and top-p 1.0. For each training configuration, we select the student checkpoint with the highest accuracy on a development set. We evaluate the selected checkpoint using 32 responses per problem on the six smaller benchmarks and 4 on OlympiadBench and MATH-500. We report mean sampled-response accuracy and its unweighted mean across the eight benchmarks, giving each benchmark equal weight regardless of size. For the main results, we compute all eight-benchmark means and

Table 1: **Mathematical reasoning accuracy** (%). Avg. is the unweighted mean over eight benchmarks. Distillation runs for both the 1.7B and 0.6B students use 16,164 training problems. Bold/underline indicate the best/second-best reported scores within each student block.

Method	Teacher	AIME			AMC	HMMT		Olympiad	MATH	Avg.
		2024	2025	2026	2023	Feb. 26	Nov. 25	Bench	500	
Qwen3-0.6B Student										
Initial student (S_0)	–	2.08	1.25	0.00	17.19	0.00	1.25	14.15	38.35	9.28
GRPO	–	9.17	13.23	12.08	46.56	6.25	7.08	35.15	65.60	24.39
OPD	T_B	13.33	12.50	<u>14.17</u>	51.25	8.33	**9.17**	39.93	70.60	27.41
TRD	T_0	4.58	7.92	5.94	39.61	6.72	4.69	29.63	61.60	20.09
IW-OPD	T_B	<u>15.00</u>	13.33	<u>14.17</u>	<u>53.75</u>	6.82	5.83	40.30	69.20	27.30
FastOPD	T_B	13.33	**21.67**	<u>14.17</u>	53.12	<u>9.09</u>	**9.17**	42.52	70.80	29.23
Relay-OPD	T_B	**17.50**	16.67	13.33	52.50	<u>9.09</u>	**9.17**	<u>45.30</u>	<u>73.60</u>	<u>29.64</u>
Prep-OPD	T_P	**17.50**	<u>18.33</u>	**15.00**	**57.50**	**10.61**	<u>7.50</u>	**45.48**	**74.45**	**30.80**
Qwen3-1.7B Student										
Initial student (S_0)	–	9.58	8.54	5.42	42.97	4.92	3.12	37.15	68.40	22.51
GRPO	–	18.02	16.77	15.62	52.50	12.97	8.02	45.70	76.40	30.75
OPD	T_B	30.00	22.50	28.33	70.00	21.97	15.00	47.63	79.50	39.37
TRD	T_B	19.17	15.83	13.75	48.44	11.36	4.17	44.48	74.60	28.97
IW-OPD	T_B	29.17	23.33	27.50	**77.50**	16.67	17.50	49.48	79.60	40.09
FastOPD	T_B	<u>42.50</u>	31.15	28.44	<u>76.48</u>	22.35	**21.46**	53.41	80.55	44.54
Relay-OPD	T_B	38.33	<u>31.25</u>	<u>30.83</u>	74.69	<u>23.48</u>	<u>20.94</u>	<u>58.00</u>	<u>85.25</u>	<u>45.35</u>
Prep-OPD	T_P	**48.33**	**31.67**	**32.08**	75.62	**25.00**	20.83	**60.44**	**87.20**	**47.65**

between-method differences before rounding. Training hyperparameters are summarized in Table 3; further implementation and evaluation details are in Appendix A.

4.2 Main Results

Overall performance. Prep-OPD achieves the highest average accuracy in both student settings (Table 1), reaching 47.65% on Qwen3-1.7B and 30.80% on Qwen3-0.6B. These scores exceed those of the strongest baseline in Table 1, Relay-OPD(T_B), by 2.30 and 1.15 percentage points, respectively. Prep-OPD obtains the best or tied-best scores on six of the eight benchmarks for each student size.

On Qwen3-1.7B, the average improvements over Standard OPD and Relay-OPD(T_B) are 8.28 and 2.30 points, respectively. Prep-OPD outperforms Standard OPD on all eight benchmarks and Relay-OPD(T_B) on seven. The largest gains over Standard OPD occur on AIME 2024 and OlympiadBench, at 18.33 and 12.81 points, respectively.

Appendix A.7 (Table 7) compares Prep-OPD and Relay-OPD(T_B) across seeds 42, 43, and 44. Prep-OPD achieves higher average accuracy in all three seeds; the means of the reported run averages are 47.37% and 45.17%, respectively.

Reusing the prepared teacher across students. The 0.6B experiments reuse the same T_P trained on prefixes from Qwen3-1.7B. The improvements on Qwen3-0.6B show that the prepared teacher can also benefit a smaller student whose prefixes were not used during teacher preparation, supporting teacher reuse across the two tested student sizes.

Training efficiency. Figure 3(a) compares average benchmark accuracy against cumulative student training compute on Qwen3-1.7B, using the checkpoint selected on the development set for each method. Prep-OPD reaches 47.65% at step 36 with 16.42 GPU-hours, compared with 45.35% at 18.75 GPU-hours for Relay-OPD(T_B). This corresponds to 2.30 percentage points higher accuracy with 12.4% less student training compute. Relative to the FastOPD checkpoint in Table 1 (44.54% at 36.88 GPU-hours), Prep-OPD improves accuracy by 3.11 points with 55.5% less student training compute.

Figure 3(b) reports mean rollout lengths over the full 126-step student training runs. Prep-OPD averages 3,239 tokens, compared with 4,894 for Standard OPD and 4,924 for IW-OPD, reductions

of 33.8% and 34.2%, respectively. Its mean length is close to Relay-OPD's 3,366 tokens, while FastOPD (3,043) and TRD (3,087) produce shorter rollouts. FastOPD uses a 4,096-token rollout cap.

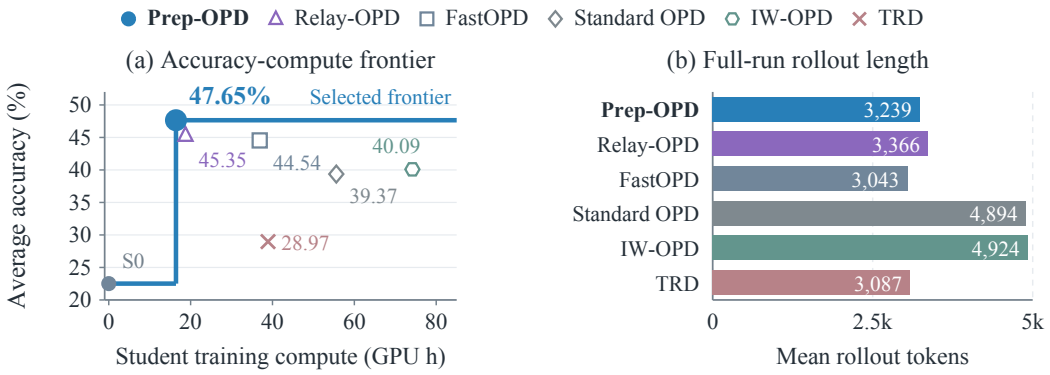


Figure 3: **Training efficiency on Qwen3-1.7B.** (a) Accuracy versus cumulative student training compute for the checkpoint selected on the development set for each distillation method. Filled markers identify Pareto-optimal points among these selected checkpoints; the staircase gives their best accuracy within each compute budget. (b) Mean rollout length over the full 126-step student training run for each method.

4.3 Teacher Continuation Capability

This subsection compares problem-start and student-prefix continuation accuracy for the original teacher T_0, the problem-start RL teacher T_B, and the prepared teacher T_P across eight benchmarks (Figure 4; Appendix A.4). All teachers continue the same frozen student prefixes, allowing direct comparison on shared inherited reasoning states. Continuation accuracy is averaged equally over prefix depths of 20%, 40%, 60%, and 80%, and overall results weight the eight benchmarks equally. The *recovery gap*, $G_T = C_T - P_T$, contrasts problem-start accuracy C_T with mean continuation accuracy P_T.

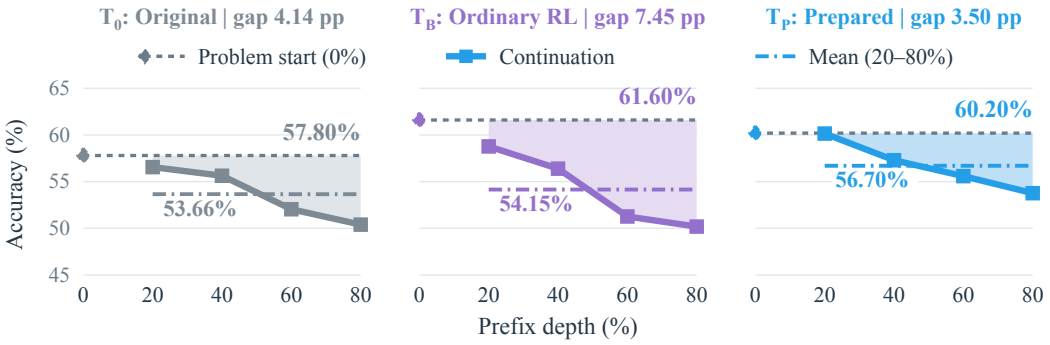


Figure 4: **Teacher continuation across prefix depths.** Accuracy at each depth weights the eight benchmarks equally, and the depth mean weights the four continuation positions equally. The 0% point denotes problem-start solving and is excluded from this mean. Panel titles report the mean recovery gap in percentage points. Shading shows the pointwise gap between problem-start and continuation accuracy, not uncertainty. Ordinary RL denotes problem-start RL.

Independent solving leaves a student-state recovery gap. T_0 achieves a mean accuracy of 57.80% from the problem start and 53.66% when continuing from student prefixes, a gap of 4.14 percentage points. This difference reflects the distinction between producing a solution independently and completing an inherited student reasoning trajectory.

Gains from problem-start RL do not fully transfer to continuation. T_B reaches 61.60% independent accuracy, improving over T_0 by 3.80 percentage points. Its continuation accuracy is 54.15%, an improvement of only 0.50 points. The mean gap between problem-start and continuation accuracy therefore widens from 4.14 to 7.45 points. These results show that gains in independent solving do not yield corresponding gains in continuation from student prefixes.

Teacher preparation improves continuation accuracy on student prefixes. T_P reaches a mean continuation accuracy of 56.70%, exceeding T_B by 2.55 percentage points. Their independent accuracies are 60.20% and 61.60%, respectively, while the mean recovery gap narrows from 7.45 to 3.50 points. The prepared teacher achieves higher continuation accuracy despite lower independent accuracy, indicating that training on student prefixes helps the teacher continue student reasoning. The benchmark-level results in Table 5 show higher mean continuation accuracy for T_P than T_B on all eight benchmarks.

Continuation across prefix depths. At prefix depths of 20%, 40%, 60%, and 80%, T_P exceeds T_B in mean continuation accuracy by 1.38, 0.90, 4.32, and 3.58 percentage points, respectively. Its advantage spans all four positions, with larger gains at the two deeper positions, where more student-generated reasoning is already fixed.

4.4 From Teacher Capability to Student Learning

Building on the continuation gains in Section 4.3, we test whether the prepared teacher improves student learning with and without handoffs, and how its generation and supervision roles affect the resulting student.

Table 2: **Ablations of teacher preparation, handoff, and teacher roles.** Results are Qwen3-1.7B student accuracies (%), averaged over the eight benchmarks with equal weights.

(a) Teacher preparation and handoff

Teacher	No handoff	With handoff	Δ handoff (pp)
T_0	39.96	45.33	+5.36
T_B	39.37	45.35	+5.98
T_P	**41.26**	**47.65**	+6.39

(b) Generation and supervision

Generation teacher	Supervision teacher	Avg. (%)
T_B	T_B	45.35
T_P	T_B	46.67
T_P	T_P	**47.65**

No handoff uses Standard OPD; all rows in (b) enable handoff. Generation covers handoff decisions and teacher segments; supervision scores all response tokens. Differences in (a) use unrounded means.

Student gains with and without teacher handoffs. Without teacher handoffs, replacing the problem-start-RL teacher T_B with the prepared teacher T_P increases average student accuracy from 39.37% to 41.26% ($+1.90$ percentage points; Table 2(a)). Both configurations use Standard OPD, where the teacher provides only token-level supervision on student-generated trajectories. The observed gain therefore does not require teacher-generated interventions. With handoffs enabled in both configurations, the same teacher replacement improves accuracy from 45.35% to 47.65% ($+2.30$ points). These comparisons support the downstream value of teacher preparation over problem-start teacher RL in both settings: when the teacher serves solely as a supervisor and when it also participates in trajectory generation. Reading across rows, handoffs improve accuracy for every teacher, while T_P achieves the highest accuracy in both columns.

Teacher roles in generation and supervision. Table 2(b) examines the prepared teacher's value in each role. Generation includes the handoff decision and teacher segment, whereas supervision scores every realized response token. Each configuration trains its own student online and induces its own rollout distribution. Comparisons between these configurations measure conditional effects rather than independent contributions of the two roles. Holding the supervision teacher fixed at T_B, replacing the generation teacher T_B with T_P increases average student accuracy from 45.35% to 46.67% (approximately 1.32 percentage points). Holding the generation teacher fixed at T_P, replacing the supervision teacher T_B with T_P increases average student accuracy from 46.67% to 47.65% (approximately 0.98 percentage points). Together, these comparisons support the value of the prepared teacher in both trajectory generation and token-level supervision.

5 Related Work

On-policy distillation. On-policy distillation (OPD) builds on knowledge distillation, which transfers teacher distributions or generated sequences to a student (Hinton et al., 2015; Kim & Rush, 2016). For LLM post-training, OPD evaluates teacher predictions along student-generated rollouts, combining on-policy sampling with dense supervision through token-level divergence losses such as KL (Gu et al., 2024; Agarwal et al., 2024). Related distillation methods refine student-data reuse and contrastive objectives (Ko et al., 2024; 2025), interpolate teacher–student distributions (Shing et al., 2025), or use teacher-derived token advantages for preference alignment (Gao et al., 2025). By comparison, RL with verifiable rewards (RLVR) often uses sparse binary outcome rewards, which cannot distinguish among failed responses (Hübotter et al., 2026). Standard supervised fine-tuning (SFT) learns from fixed demonstrations rather than trajectories sampled by the current student.

Supervision failure on student prefixes. Teacher continuation advantages can diminish as student prefixes lengthen (Li et al., 2026). More broadly, teacher hacking shows that closer imitation of an imperfect teacher on fixed offline data can increase divergence from an oracle (Tiapkin et al., 2025). Complementary distillation methods select difficult positions and adapt temperatures using teacher–student discrepancy (Xie et al., 2026a), gate token-level losses through teacher verification of student proposals (Huang et al., 2026), or adjust forward/reverse KL contributions for individual vocabulary entries (Jung et al., 2025). At the trajectory level, bounded teacher handoffs alter the student's subsequent reasoning states (Xu et al., 2026).

Teacher adaptation and privileged supervision. Distillation with privileged context conditions teacher supervision on reference solutions, demonstrations, or environmental feedback (Zhao et al., 2026; Shenfeld et al., 2026; Hübotter et al., 2026). CGD trains students to refine flawed responses using teacher critiques available only during training (Kapusuzoglu et al., 2026). Other work jointly optimizes privileged and unprivileged policies with shared parameters (Penaloza et al., 2026). SD-Zero first trains a reward-conditioned reviser through supervised fine-tuning and then distills it on policy, whereas PAST applies teacher RL conditioned on complete failed attempts and reference solutions (He et al., 2026; Feng et al., 2026). Prep-OPD optimizes teacher continuations from fixed intermediate student prefixes using suffix-only GRPO, with reference answers used solely to compute outcome rewards. The prepared teacher is frozen during student training to provide trajectory guidance and token-level supervision.

6 Conclusion

We presented Prep-OPD, which uses reinforcement learning to prepare teachers for continuation from student-generated reasoning prefixes. The prepared teacher guides student trajectories and provides token-level supervision, improving student accuracy across eight mathematical reasoning benchmarks and two student sizes. Our results show that stronger independent problem-solving ability does not necessarily imply better continuation from student prefixes. Controlled comparisons further show that student-prefix preparation improves continuation accuracy and benefits student learning both with and without handoffs. Role comparisons support its value in both trajectory generation and token-level supervision. The same prepared teacher also improves the smaller student, supporting reuse across the two tested student sizes.

AI USE STATEMENT

We have not used generative AI tools for any of the tasks with required disclosure, and the remaining required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for polishing the writing and improving the readability of the manuscript. We also used generative AI tools to assist with literature retrieval and discovery. We have reviewed all AI-assisted work: all AI-assisted edits were checked against the underlying results by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The implementation uses a fixed stack (verl 0.9.0, vLLM 0.21.0) and a versioned data pipeline: the corpus is deduplicated to 17,188 problems with a manifest recording every removal, and prefix states carry exact token IDs with recorded cut indices and exclusion statistics. The main results use seed 42; additional student runs are reported in Appendix A.7. Teacher and student protocols are described in Section 4.1 and Appendix A, including checkpoint selection on a development set and evaluation on the eight-benchmark suite. Evaluation uses the released benchmark runner of the base repository. We will release code, configurations, manifests, and the prefix-state construction tooling in an anonymous repository at submission.

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*, pp. 21246–21263, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/5be69a584901a26c521c2b51e40a4c20-Abstract-Conference.html.

Yangyang Feng, Zhuoyan Feng, and Junlan Chen. PAST: Privileged adaptation from complete student trajectories for on-policy self-distillation. *arXiv preprint arXiv:2608.08726*, 2026. URL https://arxiv.org/abs/2608.08726.

Yuqian Fu, Haohuan Huang, Kaiwen Jiang, Jiacai Liu, Zhuo Jiang, Yuanheng Zhu, and Dongbin Zhao. Revisiting On-Policy Distillation: Empirical Failure Modes and Simple Fixes. *arXiv preprint arXiv:2603.25562*, 2026. URL https://arxiv.org/abs/2603.25562.

Shiping Gao, Fanqi Wan, Jiajian Guo, Xiaojun Quan, and Qifan Wang. Advantage-guided distillation for preference alignment in small language models. In *International Conference on Learning Representations*, pp. 18916–18938, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/2f891d026c7ba978168621842bc6fe73-Abstract-Conference.html.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge Distillation of Large Language Models. In *International Conference on Learning Representations*, pp. 32694–32717, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/8ac015d409635f196f9e3e9dcfb9a94e-Abstract-Conference.html.

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. URL https://icml.cc/virtual/2026/78162. Presented at the ICML 2026 Workshop on RL from World Feedback (RLxF).

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. URL https://arxiv.org/abs/1503.02531. Presented at the NIPS 2014 Deep Learning and Representation Learning Workshop.

Haiduo Huang, Jiangcheng Song, Yadong Zhang, and Pengju Ren. SelecTKD: Selective token-weighted knowledge distillation for LLMs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19287–19297, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Huang_SelecTKD_Selective_Token-Weighted_Knowledge_Distillation_for_LLMs_CVPR_2026_paper.html.

Jonas Hübotter, Frederike Lübeck, Lejs 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 *International Conference on Machine Learning (ICML)*, 2026. URL https://icml.cc/virtual/2026/poster/64121.

Ijun Jang, Jewon Yeom, Juan Yeo, Hyunggyu Lim, and Taesup Kim. Stable On-Policy Distillation through Adaptive Target Reformulation. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 42217–42227, San Diego, California, United States, 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-acl.2094. URL https://aclanthology.org/2026.findings-acl.2094/.

Li Jiang, Haoran Xu, Yichuan Ding, and Amy Zhang. Trajectory-Refined Distillation. *arXiv preprint arXiv:2606.08432*, 2026. doi: 10.48550/arXiv.2606.08432. URL https://arxiv.org/abs/2606.08432.

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 *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, 2026. URL https://icml.cc/virtual/2026/poster/64855.

Seongryong Jung, Suwan Yoon, DongGeon Kim, and Hwanhee Lee. ToDi: Token-wise distillation via fine-grained divergence control. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 8078–8091, Suzhou, China, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.409. URL https://aclanthology.org/2025.emnlp-main.409/.

Berkcan Kapusuzoglu, Supriyo Chakraborty, Zain Sarwar, Chia-Hsuan Lee, and Sambit Sahu. Critique-guided distillation for robust reasoning via refinement. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/64994.

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, Austin, Texas, 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1139. URL https://aclanthology.org/D16-1139/.

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*, volume 235 of *Proceedings of Machine Learning Research*, pp. 24872–24895. PMLR, 2024. URL https://proceedings.mlr.press/v235/ko24c.html.

Jongwoo Ko, Tianyi Chen, Sungnyun Kim, Tianyu Ding, Luming Liang, Ilya Zharkov, and Se-Young Yun. DistiLLM-2: A contrastive approach boosts the distillation of LLMs. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 31044–31062. PMLR, 2025. URL https://proceedings.mlr.press/v267/ko25a.html.

Jahyun Koo, Yerin Hwang, Yongil Kim, Taegwan Kang, Hyunkyung Bae, and Kyomin Jung. SWITCH: Studying with teacher for knowledge distillation of large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 3733–3746, Albuquerque, New Mexico, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-naacl.206. URL https://aclanthology.org/2025.findings-naacl.206/.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huan ang Gao, Wenkai Yang, Zhiyuan Liu, and Ning Ding. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026. URL https://openreview.net/forum?id=aaDDXXDGXB. Presented at the ICML 2026 Workshop on Foundations of Deep Generative Models (FoGen); non-archival.

Alexander Lin, Jeremy Wohlwend, Howard Chen, and Tao Lei. Autoregressive Knowledge Distillation through Imitation Learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6121–6133, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.494. URL https://aclanthology.org/2020.emnlp-main.494/.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 20275–20321, Suzhou, China, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.1025. URL https://aclanthology.org/2025.emnlp-main.1025/.

Emiliano Penaloza, Dheeraj Vattikonda, Nicolas Gontier, Alexandre Lacoste, Laurent Charlin, and Massimo Caccia. Privileged information distillation for language models. In *International Conference on Machine Learning (ICML)*, 2026. URL https://icml.cc/virtual/2026/poster/62658.

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

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. In *International Conference on Machine Learning (ICML)*, 2026. URL https://icml.cc/virtual/2026/poster/61434.

Makoto Shing, Kou Misaki, Han Bao, Sho Yokoi, and Takuya Akiba. TAID: Temporally adaptive interpolated distillation for efficient knowledge transfer in language models. In *International Conference on Learning Representations*, pp. 92233–92256, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/e664650506f1cf2b4696df892147c06e-Abstract-Conference.html.

Daniil Tiapkin, Daniele Calandriello, Johan Ferret, Sarah Perrin, Nino Vieillard, Alexandre Rame, and Mathieu Blondel. On teacher hacking in language model distillation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 59552–59569. PMLR, 2025. URL https://proceedings.mlr.press/v267/tiapkin25a.html.

Xurong Xie, Zhucun Xue, Jiafu Wu, Jian Li, Yabiao Wang, Xiaobin Hu, Yong Liu, and Jiangning Zhang. LLM-oriented token-adaptive knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 34070–34078, 2026a. doi: 10.1609/aaai.v40i40.40701. URL https://ojs.aaai.org/index.php/AAAI/article/view/40701.

Yan Xie, Sijie Zhu, Tiansheng Wen, Bo Chen, and Yifei Wang. On the Position Bias of On-Policy Distillation. *arXiv preprint arXiv:2606.22600*, 2026b. URL https://arxiv.org/abs/2606.22600.

Haolei Xu, Xiaowen Xu, Haiwen Hong, Zixuan Ni, Hongxing Li, Yiwen Qiu, Weiming Lu, and Yongliang Shen. Pass the Baton: Trajectory-Relayed On-Policy Distillation. *arXiv preprint arXiv:2607.26057*, 2026. doi: 10.48550/arXiv.2607.26057. URL https://arxiv.org/abs/2607.26057.

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 *International Conference on Learning Representations*, pp. 64616–64646, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/a2747a3844ca1e4667fbff3f558eb39b-Abstract-Conference.html.

An Yang et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. In *Advances in Neural Information Processing Systems*, volume 38, pp. 113222–113244. Curran Associates, Inc., 2025. doi: 10.52202/085713-3775. URL https://proceedings.nips.cc/paper_files/paper/2025/hash/a4277440d50f1f15d2cb4c14f7e0c0d2-Abstract-Conference.html.

Dongxu Zhang, Zhichao Yang, Sepehr Janghorbani, Jun Han, Andrew Ressler, II, Qian Qian, Gregory D Lyng, Sanjit Singh Batra, and Robert E. Tillman. Fast and Effective On-Policy Distillation from Reasoning Prefixes. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 25553–25569, San Diego, California, United States, 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-acl.1276. URL https://aclanthology.org/2026.findings-acl.1276/.

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 *International Conference on Machine Learning (ICML)*, 2026. URL https://icml.cc/virtual/2026/poster/64784.

Yongchao Zhou, Kaifeng Lyu, Ankit Singh Rawat, Aditya Krishna Menon, Afshin Rostamizadeh, Sanjiv Kumar, Jean-François Kagy, and Rishabh Agarwal. DistillSpec: Improving speculative decoding via knowledge distillation. In *International Conference on Learning Representations*, pp. 32011–32050, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/8766fbc68e1ed1cdef712ce273e0a363-Abstract-Conference.html.

Ziheng Zhou, Jiaqi Li, Huacong Tang, Ying Nian Wu, and Demetri Terzopoulos. Less is More: Early Stopping Rollout for On-Policy Distillation. *arXiv preprint arXiv:2605.27028*, 2026. doi: 10.48550/arXiv.2605.27028. URL https://arxiv.org/abs/2605.27028.

A Implementation and Evaluation Details

A.1 Continuation-task construction and teacher optimization

Models and input representation. We use the unmodified `Qwen/Qwen3-1.7B` checkpoint as S_0 and `Qwen/Qwen3-4B-Instruct-2507` as T_0. The same fixed S_0 checkpoint supplies the student-generated reasoning prefixes and initializes the trainable student S_θ; the trainable teacher T_ϕ is initialized from T_0. The system message is "Please reason step by step, and put your final answer within `\boxed{}`." The user message contains the problem. We render these messages with the fixed student tokenizer, using `add_generation_prompt=True` and `enable_thinking=False`. Teacher continuation inputs concatenate the stored prompt and prefix token IDs directly, continuing the same unfinished assistant response without re-rendering the prefix or adding a new assistant turn. The reference answer is stored in a separate verifier field. Model and tokenizer revisions are fixed across stages, and rendered input IDs are retained with each task. Token-ID sharing requires compatible token-to-ID mappings, including the special tokens used for generation and stopping, across both models.

Prefix construction. For each training problem, we sample one response from S_0. Let L_y count response tokens preceding the first EOS; if no EOS is emitted, L_y is the length of the capped response. Eligible cut points satisfy $\lceil 0.15L_y \rceil \le k \le \lfloor 0.85L_y \rfloor$ and $1 \le k < L_y$. We exclude prefixes containing a complete, nonempty boxed answer or a termination token, and states whose prompt and prefix leave insufficient context for the teacher's continuation budget. From the remaining positions, we sample up to two uniformly without replacement. A trajectory with no eligible position contributes no continuation task. Each selected position defines a separate task (x, p, a) with $p = y_{1:k}$. The construction does not consult the reference answer. The retained task records define the fixed empirical distribution q_0, with equal mass assigned to each record. Teacher continuations are sampled afresh during teacher preparation. We use seed 42.

Generation budgets. We use a common 16,384-token generation cap for initial student responses, teacher preparation, student training, and benchmark evaluation. FastOPD truncates its training rollouts at 4,096 tokens. For teacher preparation, the cap applies to the newly generated suffix z; for handoff training, it covers the full response, counting both student- and teacher-generated tokens. Context windows accommodate the conditioning input and the full generation budget. Problems are retained intact; inputs that leave insufficient room for the generation budget are excluded. In particular, the stored prompt and p must leave 16,384 tokens of context capacity for teacher continuation. We use temperature 1.0, top-p 1.0, unrestricted top-k sampling, and repetition penalty 1.0.

Outcome verification. The verifier receives the decoded assistant response $p \oplus z$ and the reference answer a separately; system and user messages are not scored. We use an ordered grading chain: the first path extracts a boxed answer and checks it against a; the second path uses the supported `The answer is:` format; the final path applies symbolic parsing and equivalence checking with `math_verify`. Each path must establish answer equivalence to yield reward 1; successful extraction alone is not sufficient. The reward is 0 if no path establishes equivalence or grading raises an exception. The same procedure applies to EOS-terminated and length-truncated outputs, using the assistant text actually generated.

Group-relative advantages. At each teacher-training iteration, we fix a rollout snapshot $T_{\bar{\phi}}$ of the current teacher. For each continuation task (x_i, p_i, a_i), this snapshot samples $G = 8$ continuations $z_{i,1}, \ldots, z_{i,G}$, with rewards $r_{i,g} = R(x_i, p_i \oplus z_{i,g}, a_i)$. Define

$$\mu_i = \frac{1}{G}\sum_{g=1}^{G} r_{i,g}, \quad \sigma_i = \sqrt{\frac{1}{G-1}\sum_{g=1}^{G}(r_{i,g} - \mu_i)^2}, \\ \widehat{A}_{i,g} = \frac{r_{i,g} - \mu_i}{\sigma_i + 10^{-6}}. \tag{10}$$

Here σ_i is the sample standard deviation within the common-prefix group. Continuations from different cut points form separate groups, even when they originate from the same student response. Zero-variance groups contribute zero policy advantage and are retained without additional resampling. We perform no subsequent cross-group or batch normalization of these advantages.

Teacher optimization. At suffix position t of continuation g for task (x_i, p_i, a_i), define

$$\rho^T_{i,g,t}(\phi) = \frac{T_\phi(z_{i,g,t} \mid x_i, p_i, z_{i,g,<t})}{T_{\bar{\phi}}(z_{i,g,t} \mid x_i, p_i, z_{i,g,<t})}. \tag{11}$$

For both teacher and student updates, write the clipped surrogate as

$$c_\epsilon(\rho, A) = \min\{\rho A, \mathrm{clip}(\rho, 1-\epsilon, 1+\epsilon)A\}\,. \tag{12}$$

For a collected teacher batch, let $m^T_{i,g,t}$ mark generated suffix tokens included in optimization. We minimize

$$\mathcal{L}_T(\phi) = -\frac{\sum_{i,g,t} m^T_{i,g,t}\, c_{\epsilon_T}\left(\rho^T_{i,g,t}(\phi), \widehat{A}_{i,g}\right)}{\sum_{i,g,t} m^T_{i,g,t}}. \tag{13}$$

Here $\epsilon_T = 0.2$, and the sums cover the global teacher batch across workers. Each included suffix token has equal weight, matching $\widehat{\mathbb{E}}_{\text{suffix}}$ in Equation 8. Prompt, inherited-prefix, and padding

positions have $m^T_{i,g,t} = 0$; generated suffix tokens, including genuine EOS, have $m^T_{i,g,t} = 1$. The student prefix conditions the continuation but contributes no token-level policy-loss term. Teacher rollout log-probabilities and group advantages remain fixed while the collected batch is optimized. The rollout snapshot is refreshed before the next collection. The configured training batch and PPO mini-batch sizes are both 128. We use 128 teacher-training iterations and no reference-policy KL term. The final checkpoint is the prepared teacher T_P, whose parameters remain fixed in inference mode throughout student distillation.

Optimization and teacher serving. Table 3 summarizes the optimizer, learning rate, batching, schedule, and method-specific settings. During student distillation, the teacher is frozen and served with vLLM, using tensor-parallel size 1, `gpu_memory_utilization`$= 0.45$, and `max_model_len`$= 32{,}768$ tokens. This context limit covers the conditioning input and generated tokens; the response generation caps remain as specified above.

Table 3: **Training hyperparameters.** Batch/mini denotes the configured training batch and PPO mini-batch sizes. The prepared teacher is denoted by T_P throughout the paper.

Configuration	LR	Optimizer	Batch/mini	Schedule	Additional settings
Teacher RL (T_B, T_P)	10^{-6}	AdamW	128/128	constant	GRPO group size $G = 8$; response cap 16,384 tokens
OPD students	10^{-6}	AdamW	128/128	constant	One PPO optimization epoch; dynamic batching
Handoff students	10^{-6}	AdamW	128/128	constant	Handoff training channel (`train.sh`)
FastOPD	10^{-6}	AdamW	128/128	constant	Training rollout cap 4,096 tokens
IW-OPD	10^{-6}	AdamW	128/128	constant	Importance-weight parameter $\gamma = 0.5$

A.2 Student distillation and baseline configurations

Handoff criterion. In the implementation, we use the following reflection-disagreement criterion for Relay-OPD handoffs (Xu et al., 2026). Let $\mathcal{R}$ be the fixed reflection-token set, $v_T(h) = \arg\max_v T_P(v \mid h)$, and $d_S(h)$ the student's proposed next draft token at the current mixed prefix h. During a student segment, while the handoff budget remains, intervention is triggered when

$$g(h) = \mathbf{1}[v_T(h) \in \mathcal{R}]\, \mathbf{1}[v_T(h) \neq d_S(h)] = 1. \tag{14}$$

Both parts of the test use the same prefix h.

Reflection-token set. The reflection words are `Wait`, `But`, `Hmm`, `Actually`, `Hold`, `However`, `Yet`, `Oh`, `Alternatively`, `No`, `Ah`, `Oops`, and `Well`. We enumerate the selected tokenizer's vocabulary, decode each token ID individually, and collect IDs matching these words, including capitalization and punctuation variants, into $\mathcal{R}$. The resulting token-ID set is fixed and reused throughout Stage 2.

Teacher handoffs. Each rollout starts with the student $S_{\bar{\theta}}$, and permits at most $M = 2$ teacher handoffs. At a trigger, T_P first emits the preferred reflection token $v_T(h)$ deterministically, then samples a continuation. Each teacher segment is limited to 256 generated tokens, including the initial reflection token, and at most three natural paragraphs. Control returns to the student as soon as the third paragraph boundary is reached or the 256-token cap is reached, whichever occurs first. These limits apply separately to each teacher segment within the full-response budget.

Scoring and update timing. Before collecting a student batch, we fix the rollout snapshot $S_{\bar{\theta}}$; T_P remains fixed throughout Stage 2. Once the mixed trajectories are collected, all actual emitted response tokens are scored under both $S_{\bar{\theta}}$ and T_P on their actual mixed prefixes, before any student parameter update, using full-vocabulary log-probabilities at temperature 1.0. For trajectory i, let $h^\tau_{i,t} = (x_i, \tau_{i,<t})$. The saved advantage is

$$\widehat{A}^P_{i,t} = \mathrm{sg}\big[\log T_P(\tau_{i,t} \mid h^\tau_{i,t}) - \log S_{\bar{\theta}}(\tau_{i,t} \mid h^\tau_{i,t})\big]. \tag{15}$$

The saved scores include teacher-generated tokens, including the reflection token emitted at a handoff, scored the same way as other response tokens. There is no additional group-relative or batch normalization of $\widehat{A}^P_{i,t}$. Scores and advantages remain fixed for one student optimization epoch on the collected batch, after which the rollout snapshot is refreshed. Only the student parameters are updated.

Student token averaging. For problem–trajectory pairs in a collected batch $\mathcal{B}$, let $m_{i,t}$ mark included response tokens. Both student-generated and teacher-generated response tokens, including genuine EOS and each handoff reflection token, have $m_{i,t} = 1$. Prompt, padding, and artificial control-token positions have $m_{i,t} = 0$. The empirical average is

$$\widehat{\mathbb{E}}_{\mathcal{B}}[f] = \frac{\sum_i \sum_t m_{i,t} f_{i,t}}{\sum_i \sum_t m_{i,t}}. \tag{16}$$

The sums cover the global batch across workers, so each included response token has equal weight; we do not first normalize each trajectory by its own length. Tokens retained when a rollout is truncated remain eligible for optimization. A token added solely to signal forced termination is excluded rather than treated as a generated EOS.

Student optimization. For every included token, define the current-to-rollout student ratio

$$\rho^S_{i,t}(\theta) = \frac{S_\theta(\tau_{i,t} \mid h^\tau_{i,t})}{S_{\bar{\theta}}(\tau_{i,t} \mid h^\tau_{i,t})}. \tag{17}$$

We apply the clipped sampled-token distillation surrogate of Relay-OPD (Xu et al., 2026) with the global token average defined above:

$$\mathcal{L}_S(\theta) = -\widehat{\mathbb{E}}_{\mathcal{B}}\left[c_{\epsilon_S}\left(\rho^S_{i,t}(\theta), \widehat{A}^P_{i,t}\right)\right], \qquad \epsilon_S = 0.2. \tag{18}$$

The denominator of $\rho^S_{i,t}$ is the stored rollout-student probability for both student and teacher tokens, so teacher-generated continuations participate directly in the student loss in addition to conditioning subsequent student generation. The same frozen T_P supplies the token-level supervision throughout the mixed trajectory. Student inference uses S_θ alone.

Table 4: Full-corpus Qwen3-1.7B distillation configurations (16,164 student-training problems). Budget is the student-training response cap.

Arm	Teacher	Channel	Budget	Role
S_0	–	–	–	untrained floor
OPD	T_B	standard	16K	main distilled baseline
TRD	T_B	offline rewrite	16K	re-run of Jiang et al. (2026)
IW-OPD	T_B	standard	16K	importance-weighted OPD
FastOPD	T_B	truncated rollouts	4K	re-run of Zhang et al. (2026)
Relay-OPD	T_B	relay	16K	equal-RL handoff baseline
Prep-OPD	T_P	relay	16K	full method
Additional configurations in Table 2(a,b)				
OPD (T_0)	T_0	standard	16K	floor cell
Relay (T_0)	T_0	relay	16K	channel-only cell
OPD (T_P)	T_P	standard	16K	teacher-only cell (no handoff)
Cross (T_P/T_B)	T_P generates, T_B scores	relay	16K	role-separation arm

A.3 Evaluation and checkpoint selection

Student training budget. Both Qwen3-1.7B and Qwen3-0.6B distillation runs train for one epoch on 16,164 problems, with a configured training batch size of 128. This dataset-level epoch is distinct from the single PPO optimization epoch applied to each collected batch in Table 3.

Checkpoint selection. We use a fixed development set of 256 problems for checkpoint selection. These problems are excluded from training. For each student-training configuration, we evaluate eight candidate checkpoints on this same development set and select the checkpoint with the highest development accuracy. The selected checkpoint is then evaluated on the eight benchmarks in Section 4.1; all eight reported benchmark scores come from that checkpoint.

Table 1 and Figures 1(b) and 3(a) use Qwen3-1.7B checkpoints at steps 90 (Standard OPD), 81 (TRD), 117 (IW-OPD and FastOPD), and 36 (Relay-OPD and Prep-OPD).

For the Qwen3-0.6B configurations in Table 1, the selected checkpoints are step 110 for GRPO, step 90 for Standard OPD and IW-OPD, step 20 for TRD, step 63 for FastOPD and Relay-OPD(T_B), and step 90 for Prep-OPD. The additional Relay-OPD(T_0) comparison in Appendix A.6 uses step 90.

For benchmark evaluation, we sample 32 responses per problem on the smaller benchmarks (AIME 2024, AIME 2025, AIME 2026, AMC 2023, HMMT February 2026, and HMMT November 2025), and 4 responses per problem on the larger benchmarks (OlympiadBench and MATH-500). These sampling counts apply to all evaluated student configurations. We report mean sampled-response accuracy for each benchmark and the unweighted mean across the eight benchmarks.

Implementation checks. Five acceptance tests were run before any training arm, all passing: exact-prefix token identity between construction and continuation (32/32 static and 32/32 clean-continuation assertions); suffix-mask correctness on real pipeline tensors; the reward parser (12 assertions and 11 observational cases covering multi-boxed answers, prefix-wrong-suffix-correct relay shapes, unit and format variants, and non-crashing behavior on empty/noisy inputs); teacher isolation (checkpoints contain student-actor weights only); and checkpoint reload parity (all 310 floating-point tensors within 3.8×10^{-6} of the base, consistent with two AdamW steps at lr 10^{-6}).

A.4 Teacher conditional-capability evaluation

Figure 4 and Table 5 report teacher problem-start and student-prefix continuation accuracy across the eight benchmarks in Section 4.1. The two RL teachers share the same initialization, training problem pool, outcome verifier, and update budget; their training condition is problem-start solving for T_B and student-prefix continuation for T_P. We construct a single evaluation manifest with seed 42 from fixed S_0 outputs. All three teachers, T_0, T_B, and T_P, share the same problem set and prefix collection. They correspond to `t0`, `tb`, and `tr`, respectively, in the evaluation configuration. The evaluation conditions are problem-start solving (clean, or 0%) and continuation from prefixes at 20%, 40%, 60%, and 80% of the fixed student response's token length. These fixed evaluation positions differ from the randomized training-prefix construction in Appendix A.1. Prefix validity checks follow that construction protocol. No teacher-specific triggers or subsequent correctness outcomes select the states.

Problem-start solving generates a full response from the problem alone; student-conditioned continuation generates a suffix from the frozen prefix, with the final answer verified over the complete prefix and suffix. The two conditions use the same decoder (temperature 1.0, top-p 1.0) and the same generated-token cap. The prefix includes computation already spent by the student, so the comparison measures conditional capability rather than equal-total-inference-cost accuracy. Both conditions for a given teacher use the same final checkpoint.

Continuation samples are first averaged within problem and depth, then over the evaluated problems within each benchmark. Each benchmark's continuation accuracy is the equal-weight mean across the four depths; overall accuracies weight the eight benchmarks equally. Independent accuracy averages over evaluated problems within each benchmark and then weights the eight benchmarks equally.

Figure 4 compares the two conditions separately for T_0, T_B, and T_P, using a common accuracy scale. The 0% point denotes problem-start solving, and a horizontal dashed line marks its accuracy. A thick curve shows continuation accuracy at 20%, 40%, 60%, and 80%; a colored dash-dot line marks the equal-weight mean across these four continuation depths, excluding 0%. Problem-start accuracy and the depth mean are labeled directly on their reference lines; each panel title reports the mean gap in percentage points. Four curve markers identify the evaluated depths. The shaded region shows the pointwise gap between the problem-start reference and continuation curve, not

a confidence interval. The area of the interpolated region is not used to calculate the mean gap. The problem-start accuracies are 57.80%, 61.60%, and 60.20%, while the continuation means are 53.66%, 54.15%, and 56.70%, yielding mean gaps of 4.14, 7.45, and 3.50 percentage points, respectively.

Table 5: **Detailed teacher continuation results.** Problem-start accuracy C_T and continuation accuracy at four student-prefix depths are reported in percent. P_T is the equal-weight mean over the four depths; $G_T = C_T - P_T$ is the recovery gap in percentage points.

			Student-prefix depth					
Benchmark	Teacher	C_T	20%	40%	60%	80%	P_T	G_T
AIME 2024	T_0	60.10	60.94	56.15	53.33	54.17	56.15	3.95
	T_B	67.92	60.42	54.79	50.10	44.90	52.55	15.37
	T_P	59.27	61.46	59.48	53.65	56.15	57.68	1.58
AIME 2025	T_0	41.04	45.31	46.46	38.23	38.13	42.03	-0.99
	T_B	53.13	44.58	41.35	35.73	38.02	39.92	13.21
	T_P	44.90	45.10	42.81	44.58	41.88	43.59	1.31
AIME 2026	T_0	48.23	40.31	43.54	38.85	31.67	38.59	9.64
	T_B	52.29	51.88	42.50	39.38	38.02	42.94	9.34
	T_P	55.83	53.23	45.52	46.56	34.90	45.05	10.78
AMC 2023	T_0	89.38	81.80	79.84	83.28	83.05	81.99	7.39
	T_B	93.67	90.00	84.38	81.72	82.34	84.61	9.06
	T_P	90.70	89.14	87.27	87.81	83.20	86.86	3.84
HMMT Feb. 2026	T_0	28.88	31.53	26.61	26.80	23.67	27.15	1.73
	T_B	30.97	28.31	31.44	26.14	25.66	27.89	3.08
	T_P	28.88	31.72	29.55	27.37	29.55	29.55	-0.67
HMMT Nov. 2025	T_0	41.25	41.35	39.58	31.04	28.23	35.05	6.20
	T_B	43.75	40.42	41.25	31.98	32.71	36.59	7.16
	T_P	43.54	41.56	39.06	34.48	32.40	36.88	6.66
OlympiadBench	T_0	64.30	65.26	66.56	63.11	64.52	64.86	-0.56
	T_B	65.48	64.67	66.41	60.67	60.26	63.00	2.48
	T_P	68.81	64.93	67.07	65.56	66.52	66.02	2.79
MATH-500	T_0	89.20	85.95	86.30	81.70	79.70	83.41	5.79
	T_B	85.60	89.90	89.00	84.40	79.55	85.71	-0.11
	T_P	89.65	94.05	87.60	84.70	85.50	87.96	1.69
Macro average	T_0	57.80	56.56	55.63	52.04	50.39	53.66	4.14
	T_B	61.60	58.77	56.39	51.26	50.18	54.15	7.45
	T_P	60.20	60.15	57.30	55.59	53.76	56.70	3.50

Macro averages weight the eight benchmarks equally. All means and gaps are computed before rounding to two decimal places. T_0: original teacher; T_B: problem-start RL teacher; T_P: prepared teacher.

A.5 COMPUTE ACCOUNTING

Figure 3(a) reports cumulative student training compute through each selected checkpoint, derived from logged student-step times and an allocation of eight GPUs. Prefix generation for teacher preparation and teacher RL are separate stages and are excluded from this student-training measure. The selected Prep-OPD, Relay-OPD, and FastOPD checkpoints use 16.42, 18.75, and 36.88 GPU-hours, respectively. Figure 3(b) averages rollout length over the full 126-step student training runs, with means of $3{,}239$, $3{,}366$, $3{,}043$, $4{,}894$, $4{,}924$, and $3{,}087$ tokens for Prep-OPD, Relay-OPD, FastOPD, Standard OPD, IW-OPD, and TRD, respectively.

The mixed-teacher configuration in Table 2(b) deploys T_P for generation and T_B for scoring, so its per-step cost includes both teachers' inference – the handoff engine serving T_P at handoff points and a second forward pass through T_B to score every realized token. This configuration examines conditional changes in the generation and scoring roles.

A.6 COMPARISON WITH THE ORIGINAL-TEACHER BASELINE

Relay-OPD with T_B achieves the highest average accuracy among the baselines in Table 1 at both student sizes. Table 6 compares Prep-OPD with the corresponding Relay-OPD variant using the original teacher T_0. The Prep-OPD results at both student sizes reproduce the corresponding rows of Table 1. The 0.6B Relay-OPD (T_0) results are reported here as an additional original-teacher comparison; the 1.7B Relay-OPD (T_0) results correspond to the original-teacher configuration with handoff in Table 2(a). Prep-OPD uses T_P throughout this comparison.

For Qwen3-0.6B, Prep-OPD improves average accuracy from 29.02% to 30.80%, a gain of 1.78 percentage points, and outperforms Relay-OPD (T_0) on six of eight benchmarks. For Qwen3-1.7B, average accuracy increases from 45.33% to 47.65%, a gain of 2.32 points computed before rounding, with improvements on seven of eight benchmarks. Prep-OPD therefore retains an average-accuracy advantage over the original-teacher version of Relay-OPD at both student sizes.

Table 6: **Comparison with Relay-OPD using the original teacher.** Accuracy is reported in percent; Δ is Prep-OPD minus Relay-OPD in percentage points. Bold marks the higher accuracy within each student size. Avg. weights the eight benchmarks equally. Differences are computed before rounding.

	Qwen3-0.6B			Qwen3-1.7B		
Benchmark	Relay (T_0)	Prep (T_P)	Δ	Relay (T_0)	Prep (T_P)	Δ
AIME 2024	13.33	**17.50**	$+4.17$	43.65	**48.33**	$+4.69$
AIME 2025	14.17	**18.33**	$+4.17$	28.33	**31.67**	$+3.33$
AIME 2026	**18.33**	15.00	-3.33	31.15	**32.08**	$+0.94$
AMC 2023	50.62	**57.50**	$+6.88$	72.50	**75.62**	$+3.12$
HMMT Feb. 2026	8.33	**10.61**	$+2.27$	23.77	**25.00**	$+1.23$
HMMT Nov. 2025	**10.83**	7.50	-3.33	**21.67**	20.83	-0.83
OlympiadBench	44.07	**45.48**	$+1.41$	57.26	**60.44**	$+3.19$
MATH-500	72.45	**74.45**	$+2.00$	84.30	**87.20**	$+2.90$
Avg.	29.02	**30.80**	$+1.78$	45.33	**47.65**	$+2.32$

A.7 COMPARISON ACROSS RANDOM SEEDS

We compare Prep-OPD with Relay-OPD(T_B), the strongest baseline in Table 1, on Qwen3-1.7B using seeds 42, 43, and 44. Table 7 combines the original seed-42 results with the two additional runs for each method. Each row reports the eight benchmark accuracies and their unweighted mean.

Table 7: **Comparison across random seeds on Qwen3-1.7B.** Accuracy is reported in percent. The seed-42 rows reproduce Table 1; seeds 43 and 44 are additional runs. Avg. is the unweighted mean over the eight benchmarks, computed before rounding the benchmark entries.

Method	**Seed**	**AIME**			**AMC**	**HMMT**		**Olympiad**	**MATH**	**Avg.**
		2024	2025	2026	2023	Feb. 26	Nov. 25	Bench	500	
Prep-OPD (T_P)	42	48.33	31.67	32.08	75.62	25.00	20.83	60.44	87.20	47.65
	43	45.83	30.83	34.17	75.00	23.48	20.83	59.93	86.30	47.05
	44	43.33	31.67	37.50	73.75	26.52	23.33	58.30	85.00	47.42
Relay-OPD (T_B)	42	38.33	31.25	30.83	74.69	23.48	20.94	58.00	85.25	45.35
	43	40.00	29.58	29.58	74.38	23.48	20.83	56.67	84.45	44.87
	44	40.62	29.79	31.04	75.00	23.11	19.79	57.41	85.45	45.28

The means of the three reported run-level averages are 47.37% for Prep-OPD and 45.17% for Relay-OPD(T_B). The corresponding gains at seeds 42, 43, and 44 are 2.30, 2.18, and 2.14 percentage points. Averaging these gains before rounding gives 2.21 points. Prep-OPD has higher average accuracy in all three runs.