WHEN ALL ROLLOUTS FAIL: POLICY-COMPATIBLE JOINT DECODING FOR HARD-PROBLEM REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement Learning with Verifiable Rewards (RLVR) derives its learning signal from binary rewards via on-policy exploration and becomes a powerful paradigm for improving the reasoning of large language models. Yet it faces a learning cliff on problems beyond model's current ability: when every rollout fails, the samples reveal nothing about a correct solution, and group-relative estimators (e.g., GRPO, RLOO) collapse the update to zero advantage. Expert trajectories offer a remedy, but they are incompatible with the current policy and often lie in the student's low-probability regions, so the surviving updates demand large parameter shifts and decay into surface-level imitation rather than transferable reasoning. We argue that a useful learning signal on hard problems must be both verifiably correct and sufficiently compatible with the current policy. We propose Policy-Compatible Reinforcement Learning (**PICO**), which performs distribution-constrained joint decoding: expert data guides decoding at the token level while preserving the student's relative token probabilities, and a dynamic top-k mechanism confines the sampling distribution under a finite divergence budget, yielding trajectories that are verified and policy-compatible by construction. These trajectories are optimized jointly with on-policy rollouts under a mixed-behavior objective that mitigates their distribution mismatch via importance sampling. Experiments across reasoning diverse datasets and agentic benchmarks demonstrate that PICO consistently outperforms standard RLVR and prior off-policy baselines.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) has emerged as a prominent paradigm for post-training large language models (LLMs)(Guo et al., 2025; Jaech et al., 2024; Team et al., 2025), demonstrating strong effectiveness in enhancing reasoning capabilities across domains such as mathematics, coding, and agentic tasks(Dong et al., 2026a; Zhang et al., 2025a). RLVR enables LLMs to learn directly from verifiable reward feedback on trajectories generated through on-policy exploration, yielding substantial and generalizable performance gains(Zhu et al., 2025; Shenfeld et al., 2026). However, this reliance on successful exploration creates a learning cliff on challenging problems beyond the model's current capabilities: when all sampled rollouts fail, the policy receives no positive reward signal(Agrawal et al., 2026; Zhang et al., 2025b; Yue et al., 2025). This issue is particularly acute for methods based on group-relative advantage estimation (e.g., GRPO)(Yu et al., 2026; Zhang et al., 2026c), as all-zero-reward groups cause the estimated advantages to collapse to zero, leaving no informative learning signal to drive futher improvement on these problems.

A natural remedy is to leverage expert data, which offer a promising way to overcome this exploration bottleneck and are increasingly used to supplement online RL training. Typically generated by human experts or stronger models, expert trajectories expose the student to high-quality reasoning paths and provide informative learning signals. Yet these trajectories are not necessarily compatible with the current policy and often fall within low-probability regions under its distribution. Existing methods utilize such trajectories in three main ways. Some use them as supervised fine-tuning (SFT) targets alongside RL training(Ma et al., 2026; Fu et al., 2026; Zhang et al., 2026b). When these trajectories lie in low-probability regions under the current policy, directly fitting them can induce excessive policy drift and result in surface-level imitation rather than transferable reasoning(Yan et al., 2025). Other approaches incorporate expert trajectories into rollout groups as off-policy samples to

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.

recover informative learning signals(Yan et al., 2025; Dong et al., 2026b). The distributional mismatch between these off-policy trajectories and on-policy rollouts can distort the resulting learning signals and destabilize optimization(Agrawal et al., 2026; Setlur et al., 2026; Shenfeld et al., 2026). A further line of work(Huang et al., 2026; Qu et al., 2026; Zhang et al., 2026a;d) leverages expert trajectories indirectly, using them to construct prefixes or hints that guide subsequent exploration and generation. Although the resulting continuations are generated by the current policy, they are conditioned on intermediate states that the policy may not be able to reach on its own from the original prompt(Ross et al., 2011; Qu et al., 2026; Agrawal et al., 2026). This creates privileged-context dependence: once the guidance is removed, the model may struggle to reproduce the full reasoning process. Collectively, these limitations reveal a fundamental trade-off: *an effective learning signal on challenging problems must come from trajectories that are both verifiably correct and sufficiently compatible with the current student policy.* Although expert data can provide verifiably correct trajectories, naively incorporating them may sacrifice policy compatibility by pulling the policy toward behaviors that lie far outside its current distribution, ultimately undermining stable and effective policy improvement.

To address this challenge, we propose Policy-Compatible Reinforcement Learning (PICO), which uses distribution-constrained joint decoding to recover positive learning signals from off-policy trajectories while explicitly controlling the induced distribution shift. Specifically, joint decoding constructs the next-token candidate set by intersecting high-probability tokens predicted by a teacher conditioned on expert data with those predicted by the student, thereby restricting sampling to candidates favored by both models. In this way, the teacher provides token-level guidance to improve rollout accuracy, while sampling according to the student's relative probabilities preserves its own preferences. To further constrain the joint sampling distribution within a finite divergence budget, we introduce a dynamic top-k mechanism that adaptively adjusts the teacher's candidate set. Finally, the resulting trajectories verified for correctness are optimized alongside on-policy rollouts through a mixed-behavior objective.

As illustrated in Figure 1, PICO activates joint decoding only for all-failure groups, preserving standard on-policy optimization whenever student exploration already yields positive rewards. Figure 4 further shows that joint decoding recovers verified, positive-reward trajectories in up to 80% of all-failure groups, thereby restoring informative learning signals where standard on-policy sampling yields none. Extensive experiments across mathematics and general reasoning benchmarks demonstrate that PICO consistently outperforms standard RLVR and existing expert-guided baselines. Further evaluations on agentic tasks establish its generalizability beyond conventional reasoning settings. Comprehensive ablation studies validate the effectiveness of its key components and confirm the importance of explicitly controlling the trade-off between rollout quality and student-policy proximity. In summary, our main contributions are threefold:

- We identify a fundamental trade-off between rollout quality and student-policy proximity when incorporating offline expert information into RLVR.
- We introduce PICO, which combines distribution-constrained joint decoding with failure-triggered mixed-behavior optimization to generate and learn from verified, policy-compatible trajectories.
- Extensive experiments on mathematics, general reasoning, and agentic tasks demonstrate the effectiveness and generalizability of PICO.

2 PRELIMINARIES

Reinforcement Learning with Verifiable Rewards. Let $\mathcal{D}$ denote a training distribution over queries $x \in \mathcal{X}$. Given a query x, the policy π_θ generates a response $y = (y_1, \ldots, y_T)$ autoregressively, where T is the response length. Let $a(y)$ denote the answer extracted from y and $a^\star(x)$ the ground-truth answer to x. The verifier assigns the binary reward $R(x, y) = \mathbf{1}\{a(y) = a^\star(x)\}$. The RLVR objective maximizes the expected verifier reward(Guo et al., 2025):

$$J(\theta) = \mathbb{E}_{x\sim\mathcal{D}}\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\left[R(x, y)\right]. \tag{1}$$

Group Relative Policy Optimization. To optimize this objective, policy-gradient methods are commonly employed(Schulman et al., 2017). Among them, Group Relative Policy Optimization

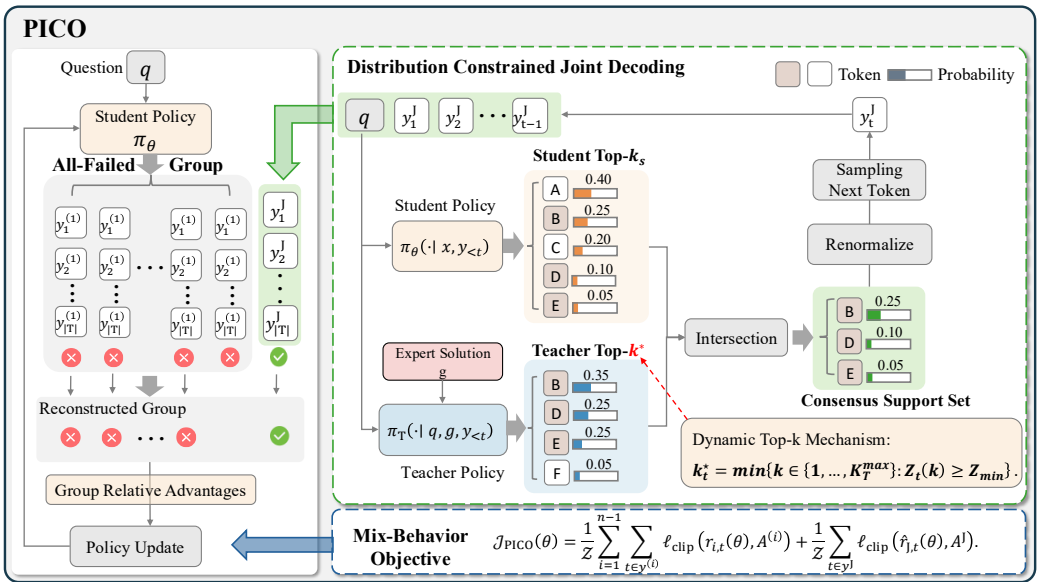


Figure 1: Overview of **PICO**. Given an all-incorrect rollout group sampled from the student policy, PICO generates a verified joint trajectory to replace one failed rollout for mixed-behavior policy optimization. The joint trajectory is generated token by token through distribution-constrained joint decoding, which intersects the teacher and student top-k candidate sets.

(GRPO) (Shao et al., 2024) is a representative approach that converts binary rewards into relative advantages within each rollout group without requiring a separately trained value model. For each query x, the behavior policy π_{old} generates a group of trajectories $\{y^{(i)}\}_{i=1}^{n}$, which are scored to obtain rewards $\{R^{(i)}\}_{i=1}^{n}$. The advantage of each trajectory is then computed by standardizing its reward within the group:

$$A^{(i)} = \frac{R^{(i)} - \text{mean}\big(\{R^{(j)}\}_{j=1}^{n}\big)}{\text{std}\big(\{R^{(j)}\}_{j=1}^{n}\big)}. \tag{2}$$

Recent work(Liu et al., 2025b) suggests that the standard deviation term in the advantage calculation may introduce bias into policy optimization. The policy is updated by optimizing the clipped, KL-regularized surrogate averaged over training problems:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{x\sim\mathcal{D}}\left[\frac{1}{\mathcal{Z}}\sum_{i=1}^{n}\sum_{t=1}^{|y^{(i)}|}\min\Big(r_{i,t}(\theta)A^{(i)}, \text{clip}(r_{i,t}(\theta), 1-\epsilon, 1+\epsilon)\,A^{(i)}\Big) - \beta\mathcal{L}_{\text{KL}}(\theta)\right]. \tag{3}$$

Here, $\mathcal{Z} = \sum_{i=1}^{n}|y^{(i)}|$ is the total number of response tokens in the rollout group, and $r_{i,t}(\theta) = \pi_\theta(y_t^{(i)} \mid x, y_{<t}^{(i)})/\pi_{\text{old}}(y_t^{(i)} \mid x, y_{<t}^{(i)})$ is the importance-sampling ratio. The clipping operation limits excessively large policy updates, while the KL term softly constrains π_θ to remain close to the reference policy π_{ref}. In practice, this KL penalty can be downweighted or omitted when larger policy shifts are desired(Yan et al., 2025; Zhang et al., 2025b).

3 Method

In this section, we detail the design and implementation of PICO. As illustrated in Figure 1 and Algorithm 1, the framework consists of two components: distribution-constrained joint decoding and mixed-behavior policy optimization. First, distribution-constrained joint decoding combines the token-level preferences of the teacher and student to generate rollouts, with a dynamic top-k mechanism adaptively balancing teacher guidance and distribution shift. Second, mixed-behavior policy optimization learns from the resulting trajectories using importance-sampling (IS) ratios to account for the mismatch between the joint-decoding policy and the student policy.

3.1 DISTRIBUTION-CONSTRAINED JOINT DECODING

To improve the reliability of sampled trajectories, we employ a teacher–student joint-decoding strategy that restricts token selection to candidates jointly favored by both models. We first construct a consensus support set and then optimize a behavior policy over this set, thereby balancing rollout quality against proximity to the student policy.

Consensus Support Set. Given a query x, privileged expert guidance g available only to the teacher, and a shared generated prefix $y_{<t}$, the next-token distributions of the student and the teacher models are defined as

$$p_t(v) = \pi_{\text{old}}(v \mid x, y_{<t}), \qquad q_t(v) = \pi_{\text{T}}(v \mid x, g, y_{<t}), \tag{4}$$

where v denotes a token in the shared vocabulary. Both models condition on the same generated prefix, while only the teacher has access to g. Thus, the privileged guidance is neither exposed to the student nor appended to its rollout context; it influences generation only through the teacher's token-level predictions.

At each decoding step, we construct the consensus support set by intersecting the top-k sets of the two models:

$$F_t = \text{TopK}(q_t, k_{\text{T}}) \cap \text{TopK}(p_t, k_{\text{S}}), \tag{5}$$

where k_{T} and k_{S} denote the teacher and student cutoffs, respectively. By retaining only tokens supported by both models, F_t filters out predictions that are implausible under either distribution. The teacher contributes privileged information through candidate selection, while the student continues to determine which of the retained tokens are compatible with its current policy.

Joint-Decoding Policy Optimization. Based on a nonempty candidate set F_t, the remaining challenge is to assign sampling probabilities within this set. The resulting behavior policy must balance two competing objectives. On the one hand, it should assign greater probability to tokens favored by the teacher so that privileged guidance can improve rollout quality. On the other hand, it should remain close to the rollout-time student distribution, thereby keeping the joint trajectory compatible to student optimization.

We quantify these objectives using the expected teacher confidence $\mathbb{E}_{v\sim\mu}[q_t(v)]$ and the reverse KL divergence $D_{\text{KL}}(\mu\|p_t)$, respectively. This yields the following joint-decoding objective:

$$\mu_t^\lambda = \arg\min_{\mu\in\Delta(F_t)} \left[D_{\text{KL}}(\mu\|p_t) - \lambda\, \mathbb{E}_{v\sim\mu}[q_t(v)] \right], \tag{6}$$

where $\Delta(F_t)$ is the set of probability distributions supported on F_t, and $\lambda \geq 0$ controls the trade-off between teacher guidance and proximity to the student policy. A smaller λ better preserves the student's relative token preferences within F_t, whereas a larger λ gives greater influence to the teacher. We describe how to obtain this target behavior policy in Section 3.2.

3.2 DYNAMIC TOP-k JOINT DECODING

Closed-Form Behavior Policy. For a nonempty support set F_t, the optimization problem in Eq. equation 6 admits the following closed-form solution:

$$\mu_t^\lambda(v) = \frac{p_t(v)\exp(\lambda q_t(v))\,\mathbf{1}\{v\in F_t\}}{C_t(\lambda)}, \qquad C_t(\lambda) = \sum_{u\in F_t} p_t(u)\exp(\lambda q_t(u)), \tag{7}$$

where $C_t(\lambda)$ is the normalizing constant. The solution incorporates teacher guidance through two complementary mechanisms: F_t imposes a hard support constraint by retaining only tokens selected by both models, while $\exp(\lambda q_t(v))$ softly reweights the student probabilities within this support. The distribution shifts induced by support restriction and soft reweighting can be separated exactly.

Proposition 1 (KL divergence decomposition)**.** *Let $Z_t = \sum_{v\in F_t} p_t(v)$ denote the student probability mass retained by F_t, and let μ_t^0 denote the policy in Eq. equation 7 at $\lambda = 0$. Then*

$$D_{\text{KL}}(\mu_t^\lambda\|p_t) = -\log Z_t + D_{\text{KL}}(\mu_t^\lambda\|\mu_t^0) \geq -\log Z_t. \tag{8}$$

The lower bound is attained at $\lambda = 0$, for which $\mu_t^\lambda = \mu_t^0$.

The first term, $-\log Z_t$, measures the shift caused by restricting the sampling support to F_t, whereas the second term measures the additional shift introduced by soft teacher reweighting. Therefore, for a fixed support set F_t, the distribution shift is minimized at $\lambda = 0$. Empirically, we find that the support constraint alone achieves sufficient rollout accuracy, making additional teacher-based reweighting unnecessary. We therefore set $\lambda = 0$ and obtain our final hard joint-decoding policy:

$$\mu_t^{\mathrm{H}}(v) = \mu_t^0(v) = \frac{p_t(v)\mathbf{1}\{v \in F_t\}}{Z_t}. \tag{9}$$

This policy preserves the student's relative token probabilities within F_t, while the teacher continues to guide generation through support selection. The derivation of Eq. equation 7 and the proof of Proposition 1 are provided in AppendixA.1.

Dynamic Top-k Selection. Under hard joint decoding, the per-step divergence is determined exactly by the retained student probability mass, $D_{\mathrm{KL}}(\mu_t^{\mathrm{H}} \| p_t) = -\log Z_t$. Therefore, imposing $Z_t \geq Z_{\min}$ guarantees $D_{\mathrm{KL}}(\mu_t^{\mathrm{H}} \| p_t) \leq -\log Z_{\min}$. To enforce this constraint adaptively, we fix the student cutoff K_{S} and, for each candidate teacher cutoff k, define

$$F_t(k) = \mathrm{TopK}(q_t, k) \cap \mathrm{TopK}(p_t, K_{\mathrm{S}}), \qquad Z_t(k) = \sum_{v \in F_t(k)} p_t(v). \tag{10}$$

We select the smallest teacher cutoff whose consensus support retains at least $Z_{\min}$ student probability mass:

$$k_t^\star = \min\left\{k \in \{1, \ldots, K_{\mathrm{T}}^{\max}\} : Z_t(k) \geq Z_{\min}\right\}. \tag{11}$$

In practice, we increase k from 1 to $K_{\mathrm{T}}^{\max}$ and stop at the first value satisfying the retained- mass constraint. We then set $F_t = F_t(k_t^\star)$ and $Z_t = Z_t(k_t^\star)$If no feasible cutoff exists, we fall back to the student policy by setting $\mu_t = p_t$ for that decoding step.

3.3 Mixed-Behavior Policy Optimization

Failure-Triggered Group Reconstruction. Following the GRPO procedure described in Section 2, we invoke joint decoding only when all n trajectories in a rollout group are incorrect. we sample one additional trajectory y^{J} using the hard joint-decoding policy. If y^{J} passes verification, it replaces one incorrect trajectory in the original group. Without loss of generality, let $y^{(n)}$ be the replaced trajectory. The resulting mixed-behavior group and its reward set are

$$\widetilde{G} = \{y^{(1)}, \ldots, y^{(n-1)}, y^{\mathrm{J}}\}, \qquad \widetilde{\mathcal{R}} = \{R^{(1)}, \ldots, R^{(n-1)}, R^{\mathrm{J}}\}. \tag{12}$$

For a successfully reconstructed group, the advantage of each trajectory is computed as

$$A^{(i)} = \widetilde{R}^{(i)} - \mathrm{mean}\left(\widetilde{\mathcal{R}}\right), \tag{13}$$

where $\widetilde{R}^{(i)} = R^{(i)}$ for $i < n$, $\widetilde{R}^{(n)} = R^{\mathrm{J}}$, and $A^{\mathrm{J}} := A^{(n)}$. The verified joint trajectory thus restores reward variation to a group that would otherwise provide no group-relative learning signal.

Mixed-Behavior Policy Optimization. The reconstructed group contains $n-1$ trajectories sampled from π_{old} and one trajectory sampled from the joint behavior policy. We optimize the student using the following clipped surrogate objective, averaged over training problems:

$$\begin{aligned}\mathcal{J}_{\mathrm{PICO}(\theta)} = \mathbb{E}_{x\sim\mathcal{D}}\Bigg[\frac{1}{\mathcal{Z}}\Bigg(&\underbrace{\sum_{i=1}^{n-1}\sum_{t=1}^{|y^{(i)}|}\min\Big(r_{i,t}(\theta)A^{(i)}, \mathrm{clip}(r_{i,t}(\theta), 1-\epsilon, 1+\epsilon)\,A^{(i)}\Big)}_{\text{student trajectories}} \\ &+\underbrace{\sum_{t=1}^{|y^{\mathrm{J}}|}\min\big(\widehat{r}_{\mathrm{J},t}(\theta)A^{\mathrm{J}}, \mathrm{clip}(\widehat{r}_{\mathrm{J},t}(\theta), 1-\epsilon, 1+\epsilon)\,A^{\mathrm{J}}\big)}_{\text{joint trajectory}}\Bigg)\Bigg],\end{aligned} \tag{14}$$

where $\mathcal{Z} = \sum_{i=1}^{n-1}|y^{(i)}| + |y^{\mathrm{J}}|$ is the total number of response tokens in the reconstructed group and $\widehat{r}_{\mathrm{J},t}(\theta)$ is the off-policy importance ratio for the joint trajectory

$$\widehat{r}_{\mathrm{J},t}(\theta) = \frac{\pi_\theta\big(y_t^{\mathrm{J}} \mid x, y_{<t}^{\mathrm{J}}\big)}{\mu_t\big(y_t^{\mathrm{J}}\big)} = Z_t\frac{\pi_\theta\big(y_t^{\mathrm{J}} \mid x, y_{<t}^{\mathrm{J}}\big)}{\pi_{\mathrm{old}}\big(y_t^{\mathrm{J}} \mid x, y_{<t}^{\mathrm{J}}\big)}. \tag{15}$$

At a fallback step, $\mu_t = p_t$ and $Z_t = 1$, so $\widehat{r}_{\mathrm{J},t}(\theta)$ reduces to the standard policy ratio. This importance ratio accounts for the token-level behavior-policy mismatch introduced by joint decoding.

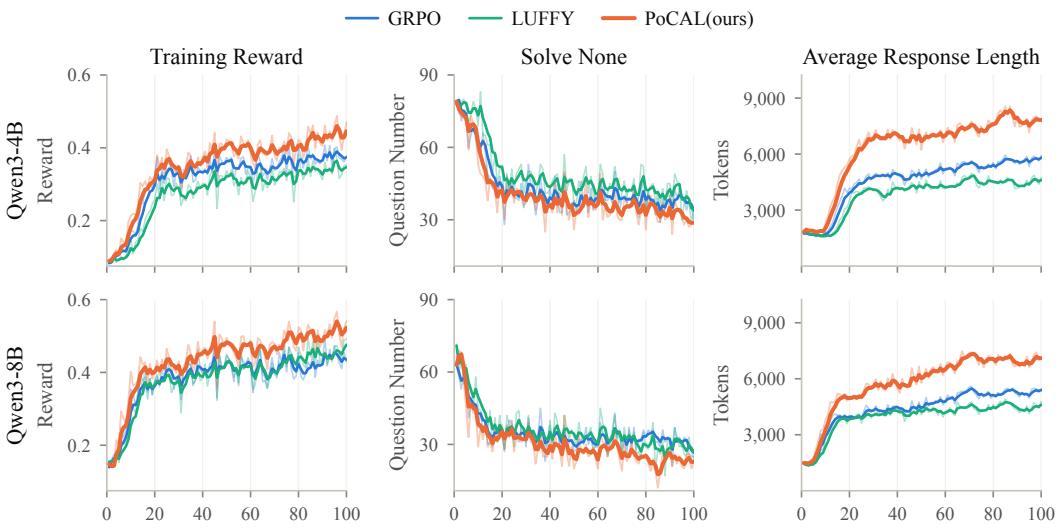


Figure 2: Training dynamics of Qwen3-4B-Base and Qwen3-8B-Base. We report the training reward, solve-none question number, and average response length throughout training.

4 Experiments

4.1 Experimental Setup

Models and Datasets. In our main experiments, we consider three student policies with different model scales and instruction-tuning configurations: Qwen3-4B-Base, Qwen3-8B-Base, and Qwen3-4B-Instruct(Yang et al., 2025). We use Qwen3-4B-Instruct specifically for the agentic-task experiments. Across all settings, Qwen3-4B-Instruct, conditioned on expert solutions, serves as the teacher model. For mathematical and general reasoning tasks, we use DAPO-Math-17K(Yu et al., 2026) as the training set, which contains a large collection of challenging problems paired with high-quality expert demonstrations. For agentic tasks, we conduct experiments on ALFWorld and ScienceWorld, two widely used interactive reasoning benchmarks.

Baselines. We compare PICO against GRPO and three representative off-policy methods , as well as a self-distillation baseline. Specifically, LUFFY(Yan et al., 2025) combines on-policy rollouts with off-policy demonstrations through mixed-policy GRPO and regularized policy shaping; Prefix-RFT(Huang et al., 2026) constructs hybrid trajectories by pairing prefixes sampled from expert demonstrations with policy-generated continuations; and ReLIFT(Ma et al., 2026) interleaves reinforcement learning with targeted online supervised fine-tuning on challenging problems. We additionally compare with RLRT(Kim et al., 2026), a self-distillation method that uses reversed teacher signals to emphasize student-preferred tokens in verified successful trajectories.

Training Details. For a fair comparison, we use the same training configuration for all RL methods. We remove the KL regularization term by setting $\beta = 0$ and omit standard-deviation normalization in the advantage computation to avoid implicitly reweighting problems according to their within-group reward variance. We fix the total rollout group size at 8 for every query. For PICO and all off-policy baselines, each mixed group consists of 7 on-policy trajectories and 1 off-policy trajectory. We use a global training batch size of 128, a PPO mini-batch size of 64, and a maximum response length of 20,480 tokens. Following DAPO, we adopt asymmetric clipping with $\epsilon_{\text{high}} = 0.28$ and $\epsilon_{\text{low}} = 0.20$. The complete training configuration is provided in Appendix C.1.

Evaluations. Following prior work, we evaluate primarily on four widely used and challenging mathematical reasoning benchmarks: AIME 2025, AIME 2026, HMMT 2026, and MATH-500. During evaluation, we set the sampling temperature to 0.6 and the maximum response length to 20,480 tokens. To reduce the effect of sampling variance, we independently sample 16 responses for each problem and report the average accuracy across these samples (avg@16). To assess out-of-

Table 1: Overall performance on in-domain and out-of-domain benchmarks, reported as avg@16. **Bold** and underlined values indicate the best and second-best results, respectively. △ denotes the absolute performance gain of our method over GRPO. Pass@k results are reported in Table 4.

Method	In-domain (%)				Out-of-domain (%)		Avg.
	AIME25	AIME26	HMMT26	MATH500	GPQA-D	HLE	
Qwen3-4B-Base							
Base	6.9	8.1	3.6	67.8	23.5	3.0	18.8
GRPO	21.7	14.4	10.0	84.6	34.3	2.2	27.9
LUFFY	22.1	18.1	14.0	84.2	36.6	3.8	29.8
Prefix-RFT	15.4	12.1	7.9	79.8	34.3	3.4	25.5
ReLIFT	10.6	10.2	6.8	76.2	31.5	3.8	23.2
RLRT	18.5	19.8	**15.9**	83.8	34.7	4.2	29.5
PICO	**23.1**	**22.3**	**15.9**	**87.6**	**37.7**	**5.2**	**32.0**
△	+1.4	+7.9	+5.9	+3.0	+3.4	+3.0	+4.1
Qwen3-8B-Base							
Base	10.2	7.5	4.4	74.8	30.1	4.4	21.9
GRPO	22.7	18.9	14.6	86.4	44.4	3.8	31.8
LUFFY	24.4	21.9	14.9	88.2	46.7	5.2	33.6
Prefix-RFT	23.9	21.0	12.1	85.8	42.6	3.4	31.5
ReLIFT	21.7	21.0	14.8	87.8	41.1	3.8	31.7
RLRT	18.8	21.9	15.9	84.4	45.7	4.6	31.9
PICO	**25.6**	**23.3**	**18.9**	**90.4**	**48.8**	**5.6**	**35.4**
△	+2.9	+4.4	+4.3	+4.0	+4.4	+1.8	+3.6

distribution generalization beyond the mathematics-focused training domain, we additionally evaluate on GPQA-Diamond and Humanity's Last Exam (HLE). Appendix C.2 provides additional information about the evaluation benchmarks.

4.2 Main Results

Performance Comparison. As shown in Table 1, PICO consistently achieves the strongest overall performance across both model scales. It improves the average score over GRPO by 4.1 percentage points with Qwen3-4B-Base and by 3.6 points with Qwen3-8B-Base, while outperforming the strongest competing method by 2.2 and 1.8 points, respectively. The gains are particularly pronounced on challenging benchmarks such as AIME26 and HMMT26, supporting our claim that PICO enables effective learning from difficult problems that frequently produce all-incorrect rollout groups. Moreover, the consistent improvements on GPQA-Diamond and HLE demonstrate that the learned reasoning capability generalizes beyond the mathematics-focused training distribution. Since the teacher is unavailable during evaluation, these gains reflect improvements internalized by the student rather than reliance on privileged guidance.

Training Dynamics. Figure 2 shows that PICO maintains higher training rewards and consistently fewer solve-none questions than GRPO and LUFFY across both model scales. While the baselines tend to plateau after the initial stage, PICO continues to improve, indicating that failure-triggered joint decoding converts otherwise uninformative all-incorrect groups into useful learning signals. PICO also produces longer responses, which, together with the higher verified rewards and lower solve-none counts, suggests more sustained reasoning on difficult problems. Overall, these results show that PICO improves difficult-problem exploration without compromising training stability. Additional analyses in Appendix B.2 and Fig.4 show that joint decoding consistently rescues questions for which all student self-sampled trajectories fail, thereby restoring effective learning signals for otherwise uninformative groups.

4.3 Agentic Tasks

We conduct the agentic-task experiments using Qwen3-4B-Instruct as both the student policy and the teacher providing token-level guidance. As shown in Table 2, PICO consistently outperforms

Table 2: Performance on agentic reasoning benchmarks based on Qwen3-4B-Instruct. **Bold** and underlined values indicate the best and second-best results, respectively. $\triangle$ denotes the absolute performance gains, in percentage points, of our method over GRPO.

Method	ALFWorld (%)		ScienceWorld (%)	Avg.
	Seen	Unseen		
Qwen3-4B-Instruct				
GRPO	80.7	71.6	50.3	67.5
LUFFY	81.4	74.6	52.1	69.4
Prefix-RFT	82.1	72.4	52.9	69.1
PICO	**88.6**	**79.1**	**55.3**	**74.3**
$\triangle$	+7.9	+7.5	+5.0	+6.8

Table 3: Ablation study of PICO on four mathematical reasoning benchmarks using Qwen3-4B-Base. "w/o Joint Decoding" directly generates responses from the teacher policy. The fixed-k variants replace the dynamic top-k mechanism with a fixed teacher candidate-set size. "w/o IS Ratio" removes the importance-sampling ratio from policy optimization. **Bold** and underlined values indicate the best and second-best results, respectively.

Method / Variant	AIME25	AIME26	HMMT26	MATH500	Avg.
Qwen3-4B-Base					
GRPO	21.7	14.4	10.0	84.6	32.7
PICO	23.1	**22.3**	15.9	**87.6**	**37.2**
w/o Joint Decoding	20.6	16.9	14.1	83.6	33.8
w/o Dynamic Top-k					
Fixed $k = 5$	**23.3**	16.9	12.7	84.2	34.3
Fixed $k = 10$	23.1	22.1	13.4	84.8	35.9
Fixed $k = 20$	21.7	18.3	**17.1**	84.0	35.3
w/o Off-Policy IS Ratio	21.7	19.4	15.2	84.8	35.3

all baselines across ALFWorld and ScienceWorld, achieving the highest average success rate of 74.3%. Compared with GRPO, PICO improves the average performance by 6.8 percentage points, demonstrating its effectiveness in interactive environments characterized by long-horizon decision-making and sparse rewards. It also surpasses the off-policy RL method LUFFY and the prefix-guided method Prefix-RFT by 4.9 and 5.2 points on average, respectively, further validating the effectiveness of token-level joint guidance for agentic reasoning.

4.4 Ablation Study

We analyze the contribution of each component of PICO. As shown in Table 3, the results highlight the importance of maintaining compatibility between guided trajectories and the student policy. Directly sampling from the teacher without joint decoding introduces the largest behavior mismatch and yields the weakest ablation result, reducing the average score from 37.2% to 33.8%. The fixed-k variants further reveal a trade-off between teacher guidance and student compatibility. Increasing k retains more student-compatible tokens but weakens the selectivity of teacher guidance, potentially reducing rollout accuracy. As long as the candidate set remains sufficiently accurate, increasing k improves learning, as evidenced by the gain from $k = 5$ to $k = 10$. However, further increasing k to 20 weakens teacher guidance and slightly reduces performance, making $k = 10$ the best fixed setting. In contrast, our dynamic top-k strategy adapts the candidate set at each decoding step according to the retained student probability mass, achieving a better balance and the highest average score of 37.2%. Finally, removing the off-policy importance-sampling ratio reduces performance to 35.3%, confirming the importance of explicitly accounting for the remaining behavior-policy mismatch during optimization.

5 Related Work

5.1 On- and Off-Policy Learning

Recent studies show that on-policy optimization is fundamentally bounded by the current policy's exploration capacity, often causing learning to stall on challenging problems (Yue et al., 2025). To overcome this exploration bottleneck, a growing line of work incorporates off-policy expert trajectories into RLVR training(Dong et al., 2026b; Liu et al., 2025a). LUFFY(Yan et al., 2025) combines expert demonstrations with on-policy rollouts through mixed-policy optimization, while ReLIFT(Ma et al., 2026) interleaves reinforcement learning with supervised fine-tuning on high-quality solutions to challenging problems. SRFT and Chord further unifies supervised learning from expert demonstrations and reinforcement learning from self-generated rollouts within a single training stage(Fu et al., 2026; Zhang et al., 2026b). However, because expert trajectories may lie in low-probability regions under the current student policy, directly optimizing on them can induce substantial distribution mismatch, potentially leading to excessive policy drift and unstable optimization (Agrawal et al., 2026; Setlur et al., 2026; Zheng et al., 2026; Yang et al., 2026). In contrast, PICO uses expert trajectories to guide distribution-constrained joint decoding, producing successful rollouts that remain compatible with the student policy.

5.2 Expert-Guided Exploration and Decoding

Rather than directly optimizing on off-policy demonstrations, another line of work uses expert information to guide rollout generation(Jiang et al., 2026; Zhang et al., 2025b; Liu et al., 2026). Prefix-RFT(Huang et al., 2026) conditions rollouts on prefixes sampled from expert demonstrations, while BREAD and StepHint use expert-derived prefixes or intermediate hints to steer exploration toward promising reasoning states(Zhang et al., 2026d;a). OC-GRPO(Agrawal et al., 2026) further applies importance sampling ratio to account for the mismatch between guided rollouts and the original unguided policy. Although these methods facilitate successful exploration, injecting guidance into the rollout context may teach the student to continue from expert-induced states without learning how to reach those states independently from the original prompt(Qu et al., 2026; Ross et al., 2011; Zhang et al., 2026c). A related line of work combines or contrasts the token distributions of models with different capabilities to steer generation toward higher-quality outputs(Liu et al., 2021; Li et al., 2023; Dekoninck et al., 2024). The effectiveness of these methods motivates us to incorporate teacher guidance at the token level. Accordingly, PICO employs distribution-constrained joint decoding to construct policy-compatible trajectories.

6 Conclusion

In this work, we introduced PICO, a policy-compatible reinforcement learning framework for overcoming the exploration bottleneck of RLVR on challenging problems. PICO leverages expert data to generate verified trajectories while controlling their deviation from the student policy, and selectively incorporates these trajectories into all-failure groups to restore informative learning signals. Extensive experiments across mathematical reasoning, general reasoning, and agentic tasks demonstrate that PICO consistently outperforms standard RLVR and representative expert-guided baselines across different model scales, without requiring teacher guidance at inference time. Ablation studies further validate the effectiveness of each component. Overall, our results highlight policy compatibility as a key principle for effectively incorporating expert knowledge into RLVR, providing a practical approach to improving models beyond the limits of unaided on-policy exploration.

References

Priyank Agrawal, Ankur Samanta, Shervin Ghasemlou, Boris Vidolov, Jalaj Bhandari, Kavosh Asadi, Daniel Jiang, and Aditya Modi. Off-context grpo: Learning to reason on hard problems using privileged information. *arXiv preprint arXiv:2607.19313*, 2026.

Jasper Dekoninck, Marc Fischer, Luca Beurer-Kellner, and Martin Vechev. Controlled text generation via language model arithmetic. In *International Conference on Learning Representations*, volume 2024, pp. 35011–35038, 2024.

Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, et al. Agentic reinforced policy optimization. In *International Conference on Learning Representations*, volume 2026, pp. 16981–17017, 2026a.

Yihong Dong, Xue Jiang, Yongding Tao, Huanyu Liu, Kechi Zhang, Lili Mou, Rongyu Cao, Yingwei Ma, Jue Chen, Binhua Li, et al. Rl-plus: Countering capability boundary collapse of llms in reinforcement learning with hybrid-policy optimization. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 43029–43053, 2026b.

Yuqian Fu, Tinghong Chen, Jiajun Chai, Xihuai Wang, Songjun Tu, Guojun Yin, Wei Lin, Qichao Zhang, Yuanheng Zhu, and Dongbin Zhao. Srft: A single-stage method with supervised and reinforcement fine-tuning for reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 102430–102459, 2026.

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.

Zeyu Huang, Tianhao Cheng, Zihan Qiu, Zili Wang, Xu Yinghui, Edoardo Ponti, and Ivan Titov. Blending supervised and reinforcement fine-tuning with prefix sampling. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=MgtyPYrC4h.

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.

Zishang Jiang, Jinyi Han, Xinyi Wang, Sihang Jiang, Zhaoqian Dai, Ma Shuguang, Fei Yu, Jiaqing Liang, Yanghua Xiao, et al. Selective expert guidance for effective and diverse exploration in reinforcement learning of llms. In *International Conference on Learning Representations*, volume 2026, pp. 62980–63006, 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.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori B Hashimoto, Luke Zettlemoyer, and Mike Lewis. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 12286–12312, 2023.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A Smith, and Yejin Choi. Dexperts: Decoding-time controlled text generation with experts and anti-experts. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 6691–6706, 2021.

Yihao Liu, Shuocheng Li, Lang Cao, Yuhang Xie, Mengyu Zhou, Haoyu Dong, Xiaojun Ma, Shi Han, and Dongmei Zhang. Superrl: Reinforcement learning with supervision to boost language model reasoning. *arXiv preprint arXiv:2506.01096*, 2025a.

Zeyuan Liu, Jeonghye Kim, Xufang Luo, Dongsheng Li, and Yuqing Yang. Exploratory memory-augmented llm agent via hybrid on-and off-policy optimization. In *International Conference on Learning Representations*, volume 2026, pp. 111799–111820, 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. *arXiv preprint arXiv:2503.20783*, 2025b.

Lu Ma, Hao Liang, Meiyi Qiang, Lexiang Tang, Xiaochen Ma, Zhen Wong, Junbo Niu, Chengyu Shen, Runming He, Yanhao Li, et al. Learning what reinforcement learning can't: Interleaved online fine-tuning for hardest questions. In *International Conference on Learning Representations*, volume 2026, pp. 80802–80821, 2026.

Yuxiao Qu, Amrith Setlur, Virginia Smith, Ruslan Salakhutdinov, and Aviral Kumar. Pope: Learning to reason on hard problems via privileged on-policy exploration. *arXiv preprint arXiv:2601.18779*, 2026.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

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

Amrith Setlur, Zijian Wang, Andrew Cohen, Paria Rashidinejad, and Sang Michael Xie. Reuse your flops: Scaling rl on hard problems by conditioning on very off-policy prefixes. *arXiv preprint arXiv:2601.18795*, 2026.

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

Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. Rl's razor: Why online reinforcement learning forgets less. In *International Conference on Learning Representations*, volume 2026, pp. 59839–59864, 2026.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. Learning to reason under off-policy guidance. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=vO8LLoNWWk.

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.

Zhihe Yang, Xufang Luo, Zilong Wang, Dongqi Han, Zhiyuan He, Dongsheng Li, and Yunjian Xu. Do not let low-probability tokens over-dominate in rl for llms. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 117072–117101, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/be0e41aa31bb4bc7e882035d82f2d781-Paper-Conference.pdf.

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, 2026.

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.

Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, Zaibin Zhang, Zelin Tan, Heng Zhou, Zhongzhi Li, Xiangyuan Xue, Yijiang Li, et al. The landscape of agentic reinforcement learning for llms: A survey. *arXiv preprint arXiv:2509.02547*, 2025a.

Kaiyi Zhang, Ang Lv, Jinpeng Li, Yongbo Wang, Feng Wang, Haoyuan Hu, and Rui Yan. Stephint: Multi-level stepwise hints enhance reinforcement learning to reason. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 37846–37864, 2026a.

Wenhao Zhang, Yuexiang Xie, Yuchang Sun, Yanxi Chen, Guoyin Wang, Yaliang Li, Bolin Ding, and Jingren Zhou. On-policy rl meets off-policy experts: Harmonizing supervised fine-tuning and reinforcement learning via dynamic weighting. In *International Conference on Learning Representations*, volume 2026, pp. 120693–120726, 2026b.

Xiaoying Zhang, Yipeng Zhang, Hao Sun, Kaituo Feng, Chaochao Lu, Chao Yang, and Helen Meng. Critique-grpo: Advancing llm reasoning with natural language and numerical feedback. *arXiv preprint arXiv:2506.03106*, 2025b.

Xichen Zhang, Sitong Wu, Yinghao Zhu, Haoru Tan, Shaozuo Yu, Ziyi He, and Jiaya Jia. Scaf-grpo: Scaffolded group relative policy optimization for enhancing llm reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 131946–131974, 2026c.

Xuechen Zhang, Zijian Huang, Yingcong Li, Chenshun Ni, Jiasi Chen, and Samet Oymak. Bread: Branched rollouts from expert anchors bridge sft & rl for reasoning. *Advances in Neural Information Processing Systems*, 38:96726–96752, 2026d.

Haizhong Zheng, Jiawei Zhao, and Beidi Chen. Prosperity before collapse: How far can off-policy rl reach with stale data on llms? In *International Conference on Learning Representations*, volume 2026, pp. 82657–82679, 2026.

Hanqing Zhu, Zhenyu Zhang, Hanxian Huang, DiJia Su, Zechun Liu, Jiawei Zhao, Igor Fedorov, Hamed Pirsiavash, Zhizhou Sha, Jinwon Lee, et al. The path not taken: Rlvr provably learns off the principals. *arXiv preprint arXiv:2511.08567*, 2025.

AI USE STATEMENT

In this work, we used generative AI tools to assist in the writing of proofs and with translation. We did not use generative AI tools to formulate mathematical claims, provide critical ingredients for proving mathematical claims, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, support qualitative or thematic data analysis, or interpret results. Generating synthetic datasets, developing theoretical models or conceptual frameworks, and cleaning or reformatting datasets were not applicable to this work. Additionally, we used generative AI tools to create or modify scientific figures or images, draft parts of the manuscript, support brainstorming, and source or search for information. We have reviewed all AI-assisted work. In particular, the authors manually verified every step of the AI-assisted proofs; checked translations against the original text to preserve their technical meaning; inspected AI-assisted figures and images for consistency with the underlying results; and carefully reviewed and revised all AI-assisted manuscript text. Ideas and information obtained through AI-assisted brainstorming or search were treated only as preliminary suggestions and were independently validated through manual literature review and consultation of the original sources. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not involve human subjects, personally identifiable or sensitive information, or the collection and release of new datasets. All experiments are conducted using publicly available models, datasets, and benchmarks for research purposes. Our method focuses on improving reinforcement learning with verifiable rewards and is not designed for high-risk or directly harmful applications.

REPRODUCIBILITY STATEMENT

To support reproducibility, Section 3 provides a complete specification of the proposed method, while Section 4.1 summarizes the main experimental setup. Appendix C provides further details on the training procedure, benchmark construction, data processing, and evaluation configurations, with the complete training hyperparameters reported in Tables 5 and 6. All prompts used for training and evaluation are provided in Appendix D.

A METHODOLOGICAL SUPPLEMENTS

A.1 THEORETICAL PROOFS

This section provides the derivations underlying the joint-decoding policy and makes explicit the guarantees induced by the retained-mass constraint. We assume that the student and teacher share a tokenizer and vocabulary $\mathcal{V}$. Since p_t is obtained from a softmax distribution, we have $p_t(v) > 0$ for every $v \in \mathcal{V}$. We also assume a fixed tie-breaking rule for top-k selection, so the selected sets are nested as k increases. The retained-mass threshold satisfies $Z_{\min} \in (0, 1]$. All token-level statements below are conditioned on a fixed query, privileged guidance, and generated prefix; we suppress this conditioning when it is clear from context.

A.1.1 OPTIMIZATION PROBLEM FORMULATION AND CLOSED-FORM SOLUTION

For a nonempty consensus set F_t, Eq. equation 6 can be written as

$$\min_{\mu \in \Delta(F_t)} \sum_{v \in F_t} \mu(v) \log \frac{\mu(v)}{p_t(v)} - \lambda \sum_{v \in F_t} \mu(v) q_t(v). \tag{16}$$

Introducing a Lagrange multiplier η for the normalization constraint gives

$$\mathcal{L}(\mu, \eta) = \sum_{v \in F_t} \mu(v) \log \frac{\mu(v)}{p_t(v)} - \lambda \sum_{v \in F_t} \mu(v) q_t(v) + \eta \left(\sum_{v \in F_t} \mu(v) - 1 \right). \tag{17}$$

At the optimum, stationarity with respect to each $\mu(v)$ yields

$$\log \frac{\mu(v)}{p_t(v)} + 1 - \lambda q_t(v) + \eta = 0, \tag{18}$$

and hence

$$\mu(v) = p_t(v) \exp(\lambda q_t(v) - 1 - \eta). \tag{19}$$

Normalizing over F_t gives

$$\mu_t^\lambda(v) = \frac{p_t(v) \exp(\lambda q_t(v)) \mathbf{1}\{v \in F_t\}}{\sum_{u \in F_t} p_t(u) \exp(\lambda q_t(u))}, \tag{20}$$

which is Eq. equation 7. The KL term is strictly convex in μ on $\Delta(F_t)$, while the teacher-confidence term is linear; therefore, this stationary point is the unique global minimizer.

A.1.2 Proof of the KL Decomposition

Let $Z_t = \sum_{v \in F_t} p_t(v)$ and define the student distribution conditioned on the consensus support as

$$\mu_t^0(v) = \frac{p_t(v) \mathbf{1}\{v \in F_t\}}{Z_t}. \tag{21}$$

For any distribution μ supported on F_t,

$$\begin{aligned} D_{\mathrm{KL}}(\mu \| p_t) &= \sum_{v \in F_t} \mu(v) \log \frac{\mu(v)}{p_t(v)} \\ &= \sum_{v \in F_t} \mu(v) \log \frac{\mu(v)}{p_t(v)/Z_t} - \log Z_t \\ &= D_{\mathrm{KL}}(\mu \| \mu_t^0) - \log Z_t. \end{aligned} \tag{22}$$

Substituting $\mu = \mu_t^\lambda$ proves Proposition 1. Since KL divergence is nonnegative, the minimum divergence over all distributions supported on F_t is $-\log Z_t$, attained uniquely by μ_t^0. Thus, once the support has been fixed, soft teacher reweighting cannot decrease the distribution shift from the student; for every $\lambda > 0$, it increases the shift strictly unless q_t is constant on F_t.

This last statement can also be seen directly. Let $m_t(\lambda) = \mathbb{E}_{v \sim \mu_t^\lambda}[q_t(v)]$. Standard properties of exponential families give

$$\frac{d}{d\lambda} m_t(\lambda) = \mathrm{Var}_{v \sim \mu_t^\lambda}[q_t(v)] \geq 0, \tag{23}$$

and

$$\frac{d}{d\lambda} D_{\mathrm{KL}}(\mu_t^\lambda \| \mu_t^0) = \lambda \, \mathrm{Var}_{v \sim \mu_t^\lambda}[q_t(v)] \geq 0. \tag{24}$$

Consequently, if q_t is nonconstant on F_t, the expected teacher confidence is strictly increasing on $[0, \infty)$, while the additional KL cost is strictly increasing on $(0, \infty)$ and has its unique minimum at $\lambda = 0$. Our hard policy selects this minimum-shift endpoint and uses the teacher only to determine the support.

Trajectory-level consequence. The per-token retained-mass constraint also controls the divergence of the complete autoregressive behavior policy. This is formalized below.

Proposition 2 (Trajectory-level KL bound)**.** *Let $\mu^{\mathrm{J}}(y \mid x, g)$ be the autoregressive joint-decoding policy and let $\pi_{\mathrm{old}}(y \mid x)$ be the rollout-time student policy. If every guided step satisfies $Z_t \geq Z_{\min}$, every infeasible step falls back to p_t, and generation has horizon $T_{\max}$, then*

$$D_{\mathrm{KL}}\big(\mu^{\mathrm{J}}(\cdot \mid x, g) \| \pi_{\mathrm{old}}(\cdot \mid x)\big) \leq T_{\max}(-\log Z_{\min}). \tag{25}$$

Here, variable-length responses are represented on the fixed horizon by treating the end-of-sequence state as absorbing and setting the two policies equal after termination. More precisely, the divergence equals the expected cumulative per-step cost $\mathbb{E}_{y \sim \mu^{\mathrm{J}}}[\sum_{t=1}^{T_{\max}} - \log Z_t]$, where $Z_t = 1$ at fallback and post-termination steps.

Proof. By the chain rule for KL divergence between autoregressive distributions,

$$D_{\mathrm{KL}}(\mu^{\mathrm{J}}\|\pi_{\mathrm{old}}) = \mathbb{E}_{y\sim\mu^{\mathrm{J}}}\left[\sum_{t=1}^{T_{\max}} D_{\mathrm{KL}}(\mu_t(\cdot \mid y_{<t})\|p_t(\cdot \mid y_{<t}))\right]$$
$$= \mathbb{E}_{y\sim\mu^{\mathrm{J}}}\left[\sum_{t=1}^{T_{\max}} -\log Z_t\right]. \tag{26}$$

The second equality follows from Eq. equation 22 at $\mu = \mu_t^0$. A fallback step has $\mu_t = p_t$ and therefore contributes zero. At every guided step, $Z_t \geq Z_{\min}$ implies $-\log Z_t \leq -\log Z_{\min}$. Summing over the horizon gives Eq. equation 25. □

A.1.3 Importance-Sampling Correction

At a fixed prefix $h_t = (x, y_{<t})$, for any sampled token $v \in F_t(h_t)$, the hard policy satisfies

$$\frac{\pi_\theta(v)}{\mu_t^{\mathrm{H}}(v)} = Z_t \frac{\pi_\theta(v)}{p_t(v)}, \tag{27}$$

which gives Eq. equation 15. The factor Z_t exactly corrects the renormalization of the student probabilities on F_t. More formally, for any bounded token-level quantity f,

$$\mathbb{E}_{v\sim\mu_t^{\mathrm{H}}}\left[\frac{\pi_\theta(v)}{\mu_t^{\mathrm{H}}(v)} f(v)\right] = \sum_{v\in F_t} \pi_\theta(v) f(v). \tag{28}$$

Thus, conditional on the prefix, the action-level correction is exact on the retained support. As with any hard-support behavior policy, it cannot recover contributions from tokens outside F_t. At data collection, when $\pi_\theta = p_t$, the magnitude of the omitted contribution is bounded by

$$\left|\mathbb{E}_{v\sim p_t}[f(v)] - \mathbb{E}_{v\sim\mu_t^{\mathrm{H}}}[Z_t f(v)]\right| \leq \|f\|_\infty (1 - Z_t) \leq \|f\|_\infty (1 - Z_{\min}). \tag{29}$$

For the rollout-time student, the retained-mass constraint therefore controls both the KL shift and the residual support truncation. Moreover, at data collection the IS factor is the token-independent constant Z_t, avoiding the additional within-step weight variance that would arise from soft teacher reweighting.

Equation equation 28 is a conditional, action-level identity; it does not convert prefixes sampled under μ^{J} into prefixes sampled under π_θ. A fully trajectory-corrected estimator would require cumulative importance ratios and target-policy absolute continuity, $\pi_\theta \ll \mu^{\mathrm{J}}$, over complete trajectories. This condition fails whenever the hard support restriction removes a token to which π_θ assigns positive probability. In addition, PPO clipping deliberately introduces bias into the surrogate. Accordingly, Eq. equation 14 should be understood as a mixed-behavior surrogate with exact per-token probability ratios on sampled actions, rather than an unbiased estimator of the original on-policy RLVR objective.

A.2 Algorithmic Procedure

Algorithm 1 gives the complete failure-triggered training procedure. The teacher is frozen throughout training, and π_{old} is held fixed during rollout collection for a policy update. The privileged guidance $g(x)$ is supplied only to the teacher; the student always evaluates the original query and the jointly generated response prefix.

B Additional Experimental Results

B.1 Pass@k Results

As shown in Table 4, PICO substantially improves Pass@16 across both model scales. Compared with GRPO, it raises the average Pass@16 by 8.8 percentage points on Qwen3-4B-Base and 10.8 points on Qwen3-8B-Base. With the 4B model, PICO achieves the best average result of 43.3%, outperforming the strongest baseline by 6.6 points. With the 8B model, it reaches 45.5% and matches the best baseline. These gains indicate that PICO improves not only average single-sample accuracy but also the likelihood of discovering at least one correct solution under repeated sampling, supporting its effectiveness in expanding successful exploration on difficult problems.

Algorithm 1 PICO: Policy-Compatible Reinforcement Learning Framework

Require: Training set $\mathcal{D} = \{(x, g_x)\}$; student π_θ; teacher π_{T}; verifier R; batch size B; group size n; K_{S}; $K_{\mathrm{T}}^{\max}$; $Z_{\min}$; $T_{\max}$; ϵ_{low}; ϵ_{high}

function JOINTDECODE($x, g_x, \pi_{\mathrm{old}}, \pi_{\mathrm{T}}$)
  $y \leftarrow ()$
  while y is not terminated and $|y| < T_{\max}$ **do**
    $p_t \leftarrow \pi_{\mathrm{old}}(\cdot \mid x, y), \quad q_t \leftarrow \pi_{\mathrm{T}}(\cdot \mid x, g_x, y)$
    $F_t(k) \leftarrow \mathrm{TopK}(q_t, k) \cap \mathrm{TopK}(p_t, K_{\mathrm{S}})$
    $Z_t(k) \leftarrow \sum_{v \in F_t(k)} p_t(v), \quad k = 1, \ldots, K_{\mathrm{T}}^{\max}$
    $\mathcal{K}_t \leftarrow \{k : Z_t(k) \geq Z_{\min}\}$
    if $\mathcal{K}_t \neq \varnothing$ **then**
      $k_t^\star \leftarrow \min \mathcal{K}_t$ ▷ *dynamic top-k*
      $F_t \leftarrow F_t(k_t^\star), \quad Z_t \leftarrow Z_t(k_t^\star)$
      $\mu_t(v) \leftarrow \dfrac{p_t(v)\mathbf{1}\{v \in F_t\}}{Z_t}$ ▷ *preserve student preferences*
    else
      $\mu_t \leftarrow p_t$
    end if
    Sample $y_t \sim \mu_t$ and append y_t to y
  end while
  return y and $\{\mu_t\}_{t=1}^{|y|}$
end function
Define $\ell(\rho, A) \leftarrow \min\{\rho A,\ \mathrm{clip}(\rho, 1 - \epsilon_{\mathrm{low}}, 1 + \epsilon_{\mathrm{high}})\, A\}$
for all $\mathcal{B} \in \mathrm{MiniBatches}(\mathcal{D}, B)$ **do**
  $\pi_{\mathrm{old}} \leftarrow \pi_\theta$
  for all $(x, g_x) \in \mathcal{B}$ **do**
    Sample $\{y_x^{(i)}\}_{i=1}^n \sim \pi_{\mathrm{old}}(\cdot \mid x)$
    $r_x^{(i)} \leftarrow R(x, y_x^{(i)}), \quad b_{x,i,t} \leftarrow \pi_{\mathrm{old}}(\cdot \mid x, y_{x,<t}^{(i)})$
    if $\sum_{i=1}^n r_x^{(i)} = 0$ **then** ▷ *trigger on an all-failed group*
      $(y^{\mathrm{J}}, \{\mu_t\}) \leftarrow$ JOINTDECODE$(x, g_x, \pi_{\mathrm{old}}, \pi_{\mathrm{T}})$
      if $R(x, y^{\mathrm{J}}) = 1$ **then**
        Replace $(y_x^{(n)}, r_x^{(n)}, b_{x,n,t})$ with $(y^{\mathrm{J}}, 1, \mu_t)$ ▷ *replace one failed rollout*
      end if
    end if
    $A_x^{(i)} \leftarrow r_x^{(i)} - \dfrac{1}{n}\sum_{j=1}^n r_x^{(j)}, \quad i = 1, \ldots, n$
  end for
  $\rho_{x,i,t}(\theta) \leftarrow \dfrac{\pi_\theta(y_{x,t}^{(i)} \mid x, y_{x,<t}^{(i)})}{b_{x,i,t}(y_{x,t}^{(i)})}$ ▷ *token-level IS correction*
  $\mathcal{Z}_{\mathcal{B}} \leftarrow \sum_{x \in \mathcal{B}} \sum_{i=1}^n |y_x^{(i)}|$
  Update θ by maximizing $\mathcal{J}_{\mathcal{B}}(\theta) = \dfrac{1}{\mathcal{Z}_{\mathcal{B}}} \displaystyle\sum_{x \in \mathcal{B}} \sum_{i=1}^{n} \sum_{t=1}^{|y_x^{(i)}|} \ell\Big(\rho_{x,i,t}(\theta), A_x^{(i)}\Big)$
end for

B.2 TRAINING CURVES

Question Rescue through Joint Decoding. Figure 4 directly illustrates how joint decoding recovers learning signals from questions that are entirely unsolved by student self-sampling. Across training, joint decoding rescues approximately 80% and 76% of these questions for Qwen3-4B and Qwen3-8B on mathematical reasoning tasks, respectively, substantially reducing the number of all-incorrect groups. It also rescues 36% of the failed questions on ALFWorld, despite the greater difficulty of long-horizon interaction. These results confirm that joint decoding enables the policy to obtain successful trajectories on problems beyond the reach of its current self-exploration, thereby converting otherwise uninformative groups into effective training signals.

Behavior-Policy Divergence. Figure 5 shows that the retained student probability mass Z_t remains consistently high throughout training, increasing from approximately 0.91 to 0.94. Correspondingly, the induced token-level KL divergence, $-\log Z_t$, decreases from roughly 0.10 to 0.06. This trend indicates that dynamic top-k joint decoding preserves most of the student probability mass and be-

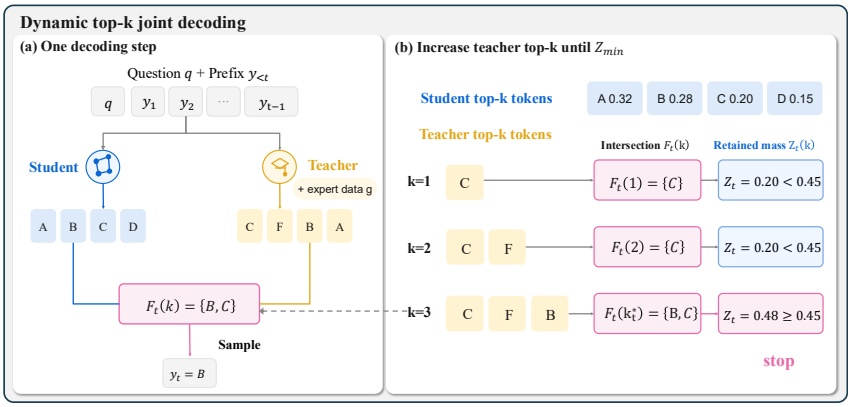


Figure 3: Illustration of dynamic top-k joint decoding. At each decoding step, the student and teacher policies produce their respective top-k token sets, whose intersection defines the candidate set $F_t(k)$. The teacher-side value of k is progressively increased until the retained student probability mass $Z_t(k)$ reaches the prescribed threshold $Z_{\min}$. The next token is then sampled from the student distribution restricted to the resulting candidate set.

Table 4: Pass@16 performance on three mathematical reasoning benchmarks. **Bold** and underlined values indicate the best and second-best results, respectively. △ denotes the absolute performance gains, in percentage points, of our method over GRPO.

Method	AIME25	AIME26	HMMT26	Avg.
Qwen3-4B-Base				
Base	26.7	16.7	18.2	20.5
GRPO	33.3	40.0	30.3	34.5
LUFFY	40.0	40.0	27.3	35.8
Prefix-RFT	40.0	36.7	24.2	33.6
ReLIFT	40.0	23.3	24.2	29.2
RLRT	36.7	40.0	33.3	36.7
PICO	46.7	50.0	33.3	43.3
△	+13.4	+10.0	+3.0	+8.8
Qwen3-8B-Base				
Base	33.3	23.3	18.2	24.9
GRPO	36.7	40.0	27.3	34.7
LUFFY	43.3	46.7	27.3	39.1
Prefix-RFT	46.7	40.0	30.3	39.0
ReLIFT	50.0	46.7	30.3	42.3
RLRT	50.0	53.3	33.3	45.5
PICO	50.0	53.3	33.3	45.5
△	+13.3	+13.3	+6.0	+10.8

comes increasingly compatible with the evolving student policy, thereby providing teacher guidance without introducing a large behavior-policy shift.

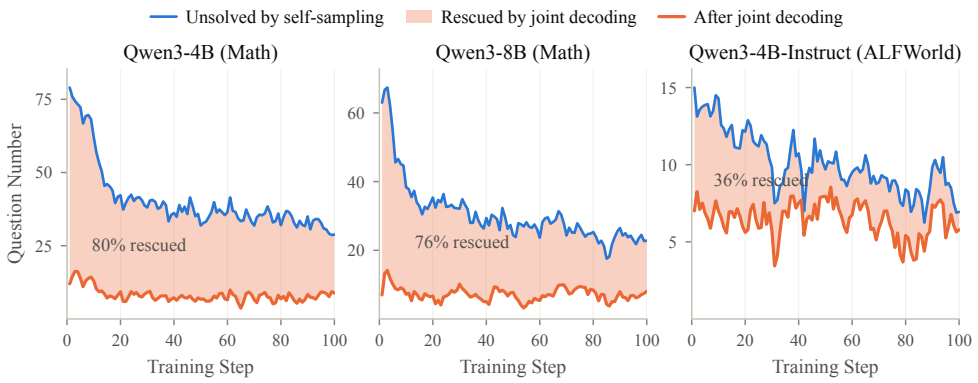


Figure 4: Dynamics of question rescue during training. The blue curves show the number of questions remaining unsolved through student self-sampling, while the orange curves show the number remaining unsolved after distribution-constrained joint decoding. The shaded regions represent questions successfully rescued by joint decoding. Results are reported for Qwen3-4B-Base and Qwen3-8B-Base on mathematical reasoning tasks, and Qwen3-4B-Instruct on ALFWorld.

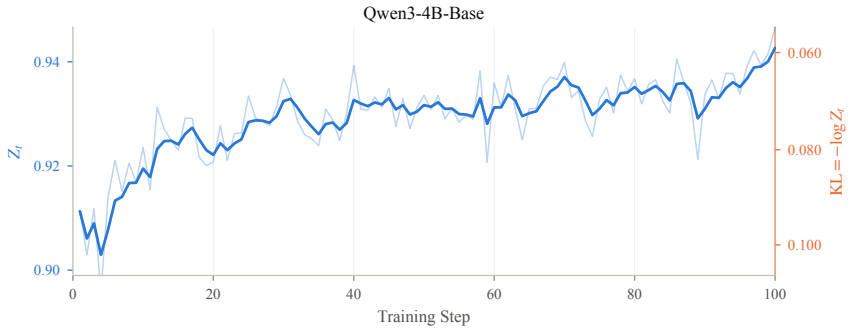


Figure 5: Evolution of the normalization term Z_t and the corresponding KL divergence, $\mathrm{KL} = -\log Z_t$, during training with Qwen3-4B-Base. The light blue curve shows the raw measurements, while the dark blue curve shows the smoothed trend.

C Experimental Details

We build all methods on top of the `verl` framework. For mathematical reasoning, we use Qwen3-4B-Base and Qwen3-8B-Base as the student policies. For a fair comparison, all reasoning methods are trained on the same DAPO-Math-17K corpus (Yu et al., 2026) using a shared hyperparameter configuration. For the agentic experiments, we train Qwen3-4B-Instruct separately on ALFWorld and ScienceWorld. Across all settings, Qwen3-4B-Instruct serves as the teacher and conditions on expert solutions to provide token-level guidance. All experiments are conducted on four NVIDIA H20 GPUs. Training either Qwen3-4B-Base or Qwen3-8B-Base on DAPO-Math-17K takes approximately 1.5–2 days, while training Qwen3-4B-Instruct for each agentic task takes approximately 1 day.

C.1 Training Details

Training Efficiency. As shown in Table 7, PICO increases the average time per training step from 28.3 to 43.6 minutes due to the additional teacher-guided decoding. This additional computation, however, improves rollout utilization from 57.2% to 91.9%, allowing substantially more sampled groups to provide effective learning signals. The utilization-per-time ratio also remains comparable to that of GRPO (2.11 vs. 2.01 percentage points per minute). Thus, PICO introduces a clear wall-clock overhead, but recovers considerably more useful training samples from each rollout batch.

Table 5: Implementation and hyperparameter configuration for mathematical reasoning experiments.

Hyperparameter	**Value**
Common (shared by all methods)	
Optimizer	AdamW
Learning rate	1e-6
Global training batch size	128
PPO mini-batch size	64
Rollout group size	8
Mixed-group composition	7 on-policy trajectories + 1 joint/off-policy trajectory
Maximum prompt length	2048
Maximum response length	20480
KL coefficient β	0
Entropy coefficient	0
PPO clipping $(\epsilon_{\text{low}}, \epsilon_{\text{high}})$	$(0.20, 0.28)$
Advantage normalization	disabled
Loss aggregation	token-level Temperature
1.0	
PICO *configuration*	
Joint-decoding policy	Hard joint policy ($\lambda = 0$)
Joint-decoding trigger	All rollouts fail
Joint-decoding response length	10240
Student cutoff K_{S}	50
Maximum teacher cutoff $K_{\text{T}}^{\max}$	20
Retained-mass threshold $Z_{\min}$	0.8

C.2 Evaluation Details

Mathematical Reasoning Benchmarks. We evaluate mathematical reasoning on AIME 2025, AIME 2026, HMMT 2026, and MATH500. AIME 2025 and AIME 2026 each contain 30 problems drawn from the corresponding AIME I and AIME II competitions. These problems require exact numerical answers and cover algebra, geometry, combinatorics, and number theory. HMMT 2026 contains 33 problems from the February 2026 Harvard–MIT Mathematics Tournament and provides a complementary evaluation of advanced competition-level mathematical reasoning. MATH-500 is a widely used subset of the MATH test set comprising 500 problems across seven mathematical subjects and a range of difficulty levels. Compared with the recent competition benchmarks, MATH-500 provides broader coverage of mathematical topics and evaluates performance on a larger and more diverse problem set. Since our reasoning models are trained on DAPO-Math-17K, we treat these three competition-mathematics benchmarks as in-domain evaluations.

Out-of-domain Reasoning Benchmarks. We evaluate cross-domain generalization on GPQA-Diamond and Humanity's Last Exam (HLE). GPQA-Diamond contains 198 expert-written, graduate-level multiple-choice questions, primarily covering physics, chemistry, and biology. The full HLE benchmark comprises 2,500 expert-level questions spanning more than 100 academic and professional subjects. We use its text-only subset, which contains 500 questions and excludes items that require multimodal inputs. These benchmarks differ substantially from the mathematics-focused training distribution and therefore assess whether the reasoning capabilities acquired during training transfer to previously unseen domains.

Agentic Benchmarks. We evaluate agentic reasoning on ALFWorld and ScienceWorld using Qwen3-4B-Instruct. ALFWorld is a text-based interactive benchmark in which an agent must complete household tasks by interpreting observations and executing sequences of admissible actions. Following the standard evaluation protocol, we evaluate on 140 task instances from the valid-seen split and 134 task instances from the valid-unseen split. ScienceWorld is an interactive text environment centered on elementary-science reasoning. It comprises 30 task types across 10 scientific

Table 6: Implementation and hyperparameter configuration for agentic experiments.

Hyperparameter	**Value**
Common optimization configuration	
Optimizer	AdamW
Learning rate	1e-6
Global training batch size	16
PPO mini-batch size	16
Rollout group size	8
Mixed-group composition	7 on-policy episodes + 1 joint/off-policy episode
Maximum prompt length	8192
Maximum response length per turn	4096
Maximum episode steps	20
KL coefficient β	0
Entropy coefficient	0
PPO clipping $(\epsilon_{\text{low}}, \epsilon_{\text{high}})$	$(0.20, 0.28)$
Advantage normalization	Disabled
Loss aggregation	Seq-mean-token-mean
Sampling temperature	1
PICO *configuration*	
Joint-decoding policy	Hard joint policy ($\lambda = 0$)
Joint-decoding trigger	All student episodes in the rollout group fail
Joint-decoding response length	2048
Student cutoff K_{S}	50
Maximum teacher cutoff $K_{\text{T}}^{\max}$	20
Retained-mass threshold $Z_{\min}$	0.8

Table 7: Comparison of training efficiency for Qwen3-4B-Base on DAPO-Math-17K. All experiments use Qwen3-4B-Base and four NVIDIA H20 GPUs. Time per step is averaged over all policy-update steps. Rollout utilization is the percentage of sampled rollout groups with non-zero group-relative advantages. Utilization/time is calculated as rollout utilization divided by average time per step and is reported in percentage points per minute (higher is better).

Method	**Time/step (min)**	**Rollout Utilization (%)**	**Utilization/time**
GRPO	28.3	57.2	2.01
PICO	43.6	91.9	2.11

topics, instantiated into 1,819 task variations. These tasks require the agent to explore the environment, manipulate objects, and complete multi-step scientific procedures. For both benchmarks, each rollout corresponds to a complete interaction episode, and we use the environment-provided task-completion signal to determine success.

Evaluation Hyperparameters. For all reasoning benchmarks, we independently sample 16 responses per problem and report both avg@16 and pass@16. We use a sampling temperature of 0.6, top-p of 1, top-k of -1, and a maximum response length of $20{,}480$ tokens. We apply the same prompt template, answer extractor, and verifier to all methods. For ALFWorld and ScienceWorld, we use a sampling temperature of 1 and a maximum of 20 interaction steps per episode. We evaluate each task instance over 8 episode(s) and report the mean episode success rate. During all evaluations, the student operates independently from the original task input without expert solutions, teacher predictions, or joint decoding.

D PROMPTS

we present the prompts used for mathematical reasoning and agentic tasks. The student prompt contains only the original task information and the interaction context available under the target policy. The teacher receives the same task context together with privileged guidance derived from off-policy data. For mathematical reasoning, this guidance is an expert solution associated with the training problem. For ALFWorld, it is a known-correct expert trajectory, which the teacher aligns with the current observation and action history to identify a suitable next action. The off-policy data is never included in the student's prompt or generated context. Instead, the teacher uses it only to produce token-level predictions during joint decoding, while the student distribution determines the relative sampling probabilities over the retained candidate tokens. This design incorporates information from successful off-policy solutions without training the student directly on their full trajectories or exposing privileged information at inference time.

D.1 MATHEMATICAL REASONING PROMPT

Mathematical Reasoning Prompt

```
You are a helpful assistant. Please reason step by step, and put
your final answer within \boxed{}.

{problem}
```

Teacher Prompt

```
You are an expert mathematician. Use the mathematical reasoning
provided in <private_guidance> to help solve the problem. Write one
complete, self-contained solution in a natural step-by-step
problem-solving style. Include the necessary equations,
intermediate conclusions, and justifications, fill in any missing
steps, and ensure that the final result is mathematically correct.
Present only the direct solution and put the final answer within
\boxed{}.

<problem>
{question}
</problem>

<private_guidance>
{prefix}
</private_guidance>

## Solution:
```

D.2 ALFWORLD PROMPTS

ALFWorld Reasoning Prompt

```
<System Prompt>

You are acting in ALFWorld TextWorld. Choose exactly one command
from the provided admissible commands on each turn. You may reason
briefly inside <think>...</think> before acting. Call the env_step
tool with that exact command. Do not output a final
natural-language answer.

<User Prompt>

### Task
{task_text}

### Current Observation
{observation}

### History Actions
{history_actions}

### Admissible Commands
{admissible_commands}

### Instructions
- Think briefly about the current state inside <think>...</think>
tags.
- Select exactly one command and execute it through the env_step
tool.
- The command must exactly match one item from Admissible Commands.
- Follow the ALFWorld TextWorld command style, such as `go to
dresser 1`, `take mug 1 from cabinet 3`, or `use desklamp 1`.
- Treat the official observation and admissible-command list as the
source of truth.
- Do not output explanations or a final natural-language answer.

### Output Format
<think>
[Brief reasoning about the current state and next command.]
</think>
<tool_call>
{{"name": "env_step", "arguments": {{"command": "[one admissible
command]"}}}}
</tool_call>
```

ALFWorld Teacher Prompt

```
<System Prompt>

You are acting in ALFWorld TextWorld. Choose exactly one command
from the provided admissible commands on each turn. You may reason
briefly inside <think>...</think> before acting. Call the env_step
tool with that exact command. Do not output a final
natural-language answer.

<User Prompt>

### Task
{task_text}

### Current Observation
{observation}

### History Actions
{history_actions}

### Expert Guidance
The following expert trajectory is a known-correct solution for the
task and is provided in full from the initial state:

{expert_trajectory}

Use the expert trajectory as private guidance. Locate the current
progress by aligning History Actions and Current Observation with
the trajectory, and identify the next action along the
corresponding solution path. Object names may differ from those in
the expert trajectory, so match entities and actions by their
semantic roles. Do not copy an expert action when it conflicts with
the current observation or is absent from Admissible Commands.

### Admissible Commands
{admissible_commands}

### Instructions
- Think briefly about the current state inside <think>...</think>
tags.
- Select exactly one command and execute it through the env_step
tool.
- The command must exactly match one item from Admissible Commands.
- Follow the ALFWorld TextWorld command style, such as `go to
dresser 1`, `take mug 1 from cabinet 3`, or `use desklamp 1`.
- Treat the official observation and admissible-command list as the
source of truth.
- Do not output explanations or a final natural-language answer.

### Output Format
<think>
[Brief reasoning about the current state and next command.]
</think>
<tool_call>
{{"name": "env_step", "arguments": {{"command": "[one admissible
command]"}}}}
</tool_call>
```