PIVOT: PROGRESSIVE INTERVENTION VIA ONLINE TEACHER CORRECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning from verifiable rewards (RLVR) is limited by sparse outcome-level feedback on hard reasoning tasks: when all rollouts in a group fail, rewards provide no learning signal. Stronger teachers can provide additional guidance, but this raises a central question: how can such signals be incorporated into RLVR to support successful rollouts while remaining learnable for the student? We distinguish two complementary mechanisms: *trajectory-level intervention*, which improves access to reward-bearing trajectories, and *token-level supervision*, which provides dense targets to guide student learning. Our controlled experiments show that either mechanism alone can remain insufficient, motivating their combination. We therefore introduce **PIVOT**, which inserts teacher-generated corrections into student rollouts and applies token-level KL supervision to these segments alongside the RLVR objective. Beyond their combination, we systematically analyze where to intervene and how to supervise, yielding a recipe for effective guidance. Across hard logic games and mathematical reasoning benchmarks, PIVOT consistently outperforms existing methods, nearly doubling mean avg@8 on logic games from 19.5% for the best-performing baseline to 35.6%.

1 INTRODUCTION

Reinforcement learning from verifiable rewards (RLVR) has become a dominant paradigm for eliciting reasoning in large language models (Shao et al., 2024; Lambert et al., 2024; Guo et al., 2025). Despite its success, recent studies suggest that RLVR primarily amplifies reasoning behaviors present in the base model, raising questions about its ability to acquire new capabilities (Yue et al., 2025; Zhao et al., 2025). This limitation is particularly pronounced on hard problems for which the initial policy rarely generates a correct solution (Yan et al., 2026; Chen et al., 2025; Qu et al., 2026). In this regime, sparse outcome-level rewards provide little information for policy improvement: groups of unsuccessful rollouts receive identical zero rewards, leaving the policy without a reward-based learning signal (Wu et al., 2026; Tao et al., 2025). These challenges have motivated the use of stronger teachers to provide guidance beyond verifier feedback. However, introducing teacher signals does not by itself ensure that the student can learn from them. This raises a central question: *how can teacher guidance support **successful** rollouts while remaining **learnable** for the student?*

Existing approaches incorporate teacher signals into RLVR through two broad mechanisms. *Trajectory-level intervention* introduces teacher-generated trajectories or segments into training, improving access to reward-bearing trajectories (Guo et al., 2025; Wen et al., 2025; Yan et al., 2025; Qu et al., 2026). However, teacher-generated trajectories may differ substantially from the student's own rollout distribution, making the teacher's behavior difficult to learn even when these trajectories are successful. *Token-level supervision*, in contrast, provides dense supervision along student-generated rollouts using the teacher's predictive distributions (Xu et al., 2025). Yet when these rollouts deviate from a valid reasoning path, local corrective signals may not support a coherent recovery. These limitations suggest that effective teacher guidance must shape both the trajectories used for training and the supervision derived from them.

We therefore introduce **P**rogressive **I**nter**v**ention via **O**nline **T**eacher Correction (PIVOT), a teacher-assisted RLVR framework that integrates teacher corrections into both rollout generation and policy optimization. PIVOT augments the rollout group with teacher-assisted trajectories, in which the teacher provides a correction based on a student-generated prefix and the student continues from the

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.

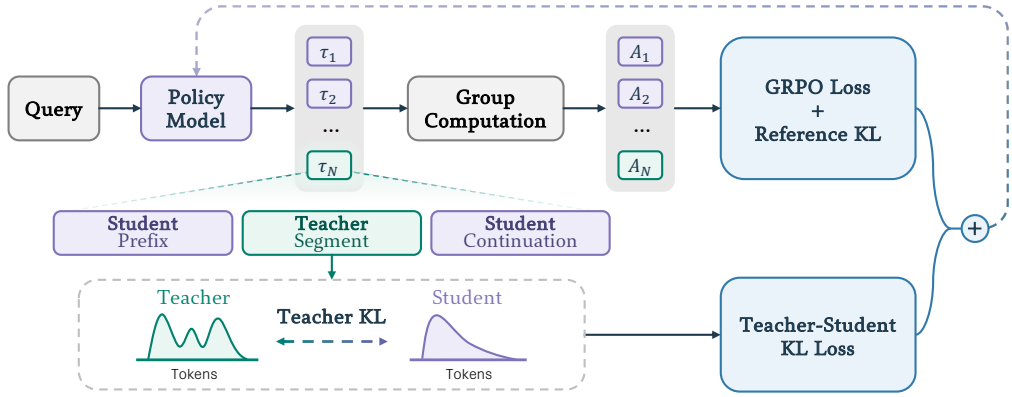

Figure 1: Overview of PIVOT. For each query, the policy model samples a group of rollouts, including teacher-assisted trajectories. In a teacher-assisted rollout, the student first generates a prefix, the teacher provides a correction segment, and the student then continues generation. The completed trajectory is optimized by the verifier-based GRPO objective, while the teacher-generated segment provides dense token-level supervision through a teacher-student KL loss.

corrected context. As illustrated in Fig. 1, verifier-based RLVR optimizes the resulting trajectories, while a teacher–student KL objective provides dense token-level supervision on the correction segments. Our controlled experiments show that this joint use of teacher signals yields stronger learning than either intervention or token-level supervision alone. We further examine how intervention placement and the KL objective affect the effectiveness of teacher guidance, informing a concrete recipe for where to intervene, how to supervise, and how much teacher guidance to apply.

We evaluate PIVOT on hard logic games from SynLogic (Liu et al., 2026) and mathematical reasoning benchmarks. On hard logic games, where RL-only training makes limited progress, PIVOT nearly doubles mean avg@8 from 19.5% for the best-performing baseline to 35.6%. On mathematical reasoning tasks, PIVOT consistently outperforms the evaluated baselines across both in-distribution and out-of-distribution benchmarks.

Our contributions are threefold:

- We identify two complementary mechanisms for teacher guidance in RLVR—*trajectory-level intervention* and *token-level supervision*—and show that either mechanism alone can remain insufficient for effective learning on hard reasoning tasks.
- We introduce PIVOT, a teacher-assisted RLVR framework that integrates teacher corrections into both rollout generation and policy optimization, and provide a systematic analysis of how teacher guidance should be incorporated into training.
- We conduct extensive experiments and show that PIVOT nearly doubles mean avg@8 over the best-performing baseline on hard logic games while consistently outperforming baselines across in-distribution and out-of-distribution mathematical reasoning benchmarks.

2 Preliminaries

Problem setup. We consider reinforcement learning with verifiable rewards (RLVR) for a student policy π_θ, with access to a teacher policy π_T during training. Given a prompt $\mathbf{q}$ drawn from the prompt distribution $\mathcal{D}$, a response $\tau = (\tau_1, \ldots, \tau_{|\tau|})$ receives a binary verifier reward $R(\mathbf{q}, \tau) \in \{0, 1\}$, with 1 indicating a correct response. The student factorizes its response distribution as $\pi_\theta(\tau \mid \mathbf{q}) = \prod_{t=1}^{|\tau|} \pi_\theta(\tau_t \mid h_t)$, where $h_t = (\mathbf{q}, \tau_{<t})$ denotes the token context. The objective is to maximize $\mathbb{E}_{\mathbf{q}\sim\mathcal{D},\, \tau\sim\pi_\theta(\cdot|\mathbf{q})}[R(\mathbf{q}, \tau)]$.

RLVR objective. Under Group Relative Policy Optimization (GRPO) (Shao et al., 2024), groups of rollouts are sampled from $\pi_{\theta_{\text{old}}}$ for each prompt, and their verifier rewards are normalized within each group to obtain advantages A_i. For a batch of N rollouts, the policy loss is averaged over all

tokens:

$$\mathcal{L}_{\mathrm{GRPO}}(\theta) = -\frac{1}{\sum_{i=1}^{N}|\tau_i|}\sum_{i=1}^{N}\sum_{t=1}^{|\tau_i|}\min\Big(r_{i,t}(\theta)A_i,\ \mathrm{clip}(r_{i,t}(\theta),1-\varepsilon,1+\varepsilon)A_i\Big),$$

Here, $|\tau_i|$ is the rollout length, $\varepsilon > 0$ is the clipping threshold, and $r_{i,t}(\theta) = \frac{\pi_\theta(\tau_{i,t}|h_{i,t})}{\pi_{\theta_{\mathrm{old}}}(\tau_{i,t}|h_{i,t})}$ is the importance ratio between the current and rollout policies. The RLVR objective additionally includes a KL penalty to a fixed reference policy π_{ref}:

$$\mathcal{L}_{\mathrm{ref}}(\theta) = \frac{1}{\sum_{i=1}^{N}|\tau_i|}\sum_{i=1}^{N}\sum_{t=1}^{|\tau_i|}\mathbb{D}_{\mathrm{KL}}(\pi_\theta(\cdot \mid h_{i,t}) \,\|\, \pi_{\mathrm{ref}}(\cdot \mid h_{i,t}))\,.$$

The RLVR loss is $\mathcal{L}_{\mathrm{GRPO}} + \beta_{\mathrm{ref}}\mathcal{L}_{\mathrm{ref}}$, with regularization weight $\beta_{\mathrm{ref}} \geq 0$.

Token-level teacher supervision. Beyond outcome-level rewards, some methods use teacher predictions to provide dense token-level supervision (Xu et al., 2025). This is commonly formulated as a forward- or reverse-KL loss between the teacher and student distributions:

$$\mathcal{L}_{\mathrm{KL}}^{\mathrm{forward}}(\theta;\mathcal{S}) = \frac{1}{|\mathcal{S}|}\sum_{(i,t)\in\mathcal{S}}\mathbb{D}_{\mathrm{KL}}(\pi_T(\cdot \mid h_{i,t}) \,\|\, \pi_\theta(\cdot \mid h_{i,t}))\,,$$

$$\mathcal{L}_{\mathrm{KL}}^{\mathrm{reverse}}(\theta;\mathcal{S}) = \frac{1}{|\mathcal{S}|}\sum_{(i,t)\in\mathcal{S}}\mathbb{D}_{\mathrm{KL}}(\pi_\theta(\cdot \mid h_{i,t}) \,\|\, \pi_T(\cdot \mid h_{i,t}))\,,$$

where $\mathcal{S}$ denotes the token positions (i,t) that receive teacher supervision.

3 RLVR through Teacher Intervention and Supervision

Teacher guidance can shape both the trajectories a student explores (i.e., *trajectory-level intervention*) and the learning signals it receives (i.e., *token-level supervision*). Existing methods typically emphasize one of these two mechanisms. Below, we examine why either mechanism alone can be insufficient for effective student learning and how combining them motivates the design of PIVOT.

3.1 Reward-Bearing Trajectories Are Not Necessarily Learnable

While the student may struggle to discover successful trajectories, *trajectory-level intervention* by a stronger teacher provides access to reward-bearing reasoning paths.

Supervised fine-tuning (SFT) on these trajectories provides a warm start for subsequent RL (Guo et al., 2025; Wen et al., 2025), while LUFFY incorporates successful teacher trajectories directly into RL training (Yan et al., 2025). POPE instead provides a teacher-generated prefix and lets the student generate the remaining continuation (Qu et al., 2026). These approaches differ in when and how teacher trajectories enter training, but all use teacher-generated trajectories or segments to improve access to successful reasoning paths.

The success of a teacher trajectory, however, does not ensure that its behavior is readily learnable by the student. Full teacher trajectories may differ substantially from the student's own rollout distribution, making them difficult imitation targets even when they receive positive rewards. Providing only a teacher prefix allows the student to generate its own continuation, but learning to continue from a supplied prefix does not necessarily teach the student to produce the teacher decisions encoded in that prefix during independent generation.

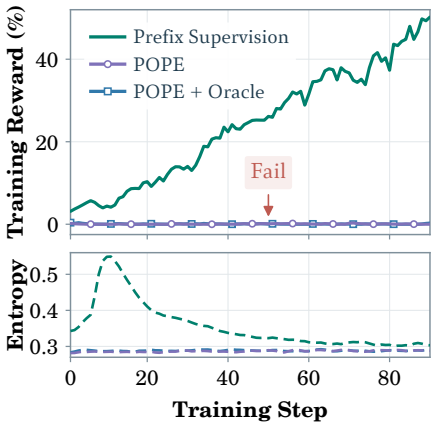

Figure 2: Prefix supervision vs. prefix-only guidance on Survo. Supervised prefixes improve rewards, while POPE-style prefixes fail to yield comparable gains.

Fig. 2 illustrates this distinction on Survo. POPE yields limited gains even with stronger teacher prefixes. We additionally consider oracle prefixes extracted from correct solutions to test whether prefix errors account for the limited gains. Even with these prefixes, POPE makes little progress, whereas adding explicit teacher supervision on the provided prefixes improves training rewards. This contrast suggests that prefix quality alone does not determine the usefulness of teacher guidance: the training objective must also enable the student to learn from the teacher-provided behavior.

3.2 Dense Supervision Does Not Guarantee Trajectory Recovery

Unlike trajectory-level intervention, *token-level supervision* introduces teacher guidance directly into the learning objective. KDRL (Xu et al., 2025), for example, combines GRPO with reverse-KL distillation along student-generated rollouts. The teacher provides dense feedback at each token, but the student still determines the reasoning path and hence the contexts on which supervision is applied.

This dependence on student-generated contexts can complicate recovery from reasoning errors. As observed in Trajectory-Refined Distillation (TRD) (Jiang et al., 2026), when a student prefix contains errors that require backtracking, the teacher may assign high probability both to continuing the student's reasoning and to initiating a correction. Under such prefix failure, the resulting bimodal teacher mixture provides conflicting targets for token-level supervision.

Fig. 3 (left) illustrates this issue on a Survo rollout trained with KDRL. After an early error, the student continues along the wrong path and the rollout ends with zero reward. At the supervised positions, teacher predictions repeatedly split between corrective tokens ("Wait") and tokens that continue the reasoning ("Let", "So"), pulling the student in two directions at once. Even when the teacher favors a correction, the remainder of the training sequence is still the student's erroneous continuation. The student thus receives local corrective signals without observing how the proposed recovery unfolds. When successful student rollouts are rare, training provides few examples of how to recover from an initial error and carry the revised reasoning through to a correct solution, limiting the ability of supervision alone to guide the student toward successful completion.

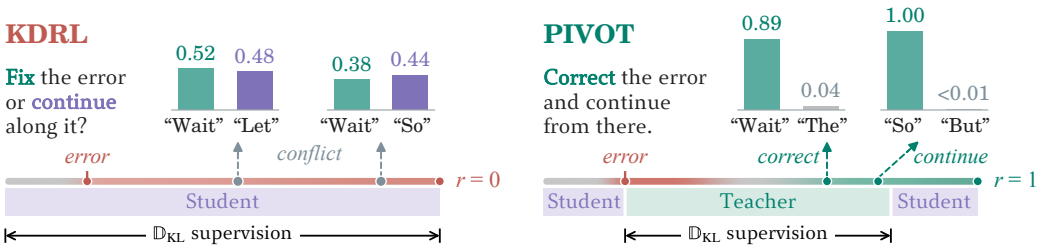

Figure 3: Dense supervision under KDRL and PIVOT on a Survo rollout; insets show the teacher's top-2 next-token probabilities at supervised positions. **Left:** on the student's failed trajectory, the teacher is split between fixing the error ("Wait") and continuing along it ("Let", "So"), giving conflicting targets. **Right:** the teacher corrects the error itself and continues from there, so its targets are unimodal whether it backtracks ("Wait") or proceeds ("So"), and the trajectory reaches the reward.

3.3 PIVOT: Integrating Trajectory Guidance and Token-Level Supervision

The preceding results suggest that the value of teacher guidance depends on both the reasoning paths it introduces and the supervision applied along those paths. For methods relying solely on *trajectory-level intervention*, Fig. 2 shows that adding a teacher-matching loss enables the student to learn more effectively from the provided prefixes. For methods that apply *token-level supervision* solely along student-generated rollouts, Fig. 3 (right) shows how teacher intervention changes the supervision: the teacher generates a recovery path that becomes part of the training sequence and leads to a successful solution, providing largely unimodal targets both when initiating a correction ("Wait") and when continuing the revised reasoning ("So").

These observations motivate PIVOT's joint use of teacher-generated segments for trajectory-level intervention and token-level supervision, allowing teacher guidance to shape both exploration and the learning objective. We next describe the rollout construction and training objective.

Teacher-assisted rollout. For each prompt $\mathbf{q}$, we sample a group of rollouts $\mathcal{G} = \mathcal{G}_S \cup \mathcal{G}_T$, comprising student-only rollouts $\mathcal{G}_S$ and teacher-assisted rollouts $\mathcal{G}_T$, with sizes G_S and G_T, respectively. Student-only rollouts are sampled directly from $\pi_{\theta_{\text{old}}}(\cdot \mid \mathbf{q})$. For teacher-assisted rollouts, the student first generates a reasoning prefix, the teacher continues from that prefix for a limited segment, and the student then resumes generation from the resulting context. Writing the resulting rollout as $\tau = (u, v, w)$, the student generates u and w, while the teacher generates v. For an intervention position $\ell_u = |u|$ and teacher segment length $\ell_v = |v|$, this gives the sampling distribution

$$p^{(\ell_u,\ell_v)}_{\theta_{\text{old}},T}(\tau \mid \mathbf{q}) = \pi_{\theta_{\text{old}}}(u \mid \mathbf{q})\, \pi_T(v \mid \mathbf{q}, u)\, \pi_{\theta_{\text{old}}}(w \mid \mathbf{q}, u, v).$$

Setting $\ell_u = 0$ places the teacher segment at the beginning of the rollout; otherwise, the teacher conditions its intervention on a student-generated prefix.

Training objective. We combine the RLVR objective with a teacher-supervision loss applied only to teacher-generated tokens. Let $\mathcal{S}_T$ denote the set of token positions belonging to the teacher-generated segments v in the training batch. The overall objective is

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{GRPO}}(\theta) + \beta_{\text{ref}}\mathcal{L}_{\text{ref}}(\theta) + \beta_{\text{T}}\mathcal{L}_{\text{KL}}(\theta; \mathcal{S}_T),$$

where $\mathcal{L}_{\text{KL}}$ is either the forward- or reverse-KL loss defined in Section 2. Together, the rollout construction and training objective define how PIVOT uses teacher guidance, but leave open where to intervene, how to supervise, and how much guidance to provide, which we study next.

4 Towards Effective Teacher Guidance: A Roadmap

We investigate where to intervene (Section 4.2), how to formulate teacher supervision (Section 4.3), and how to configure the remaining guidance settings (Section 4.4) within PIVOT. These analyses yield a practical training recipe, which we evaluate against existing methods in Section 4.5.

4.1 Experimental Setup

Tasks. We conduct experiments on hard logic games and mathematical reasoning. For logic games, we train and evaluate on the hard subsets of Survo, Minesweeper, and Kakurasu from SynLogic (Liu et al., 2026), on which the base student almost never succeeds. For mathematical reasoning, we pool problems from Polaris-53K (An et al., 2025) and the math subset of Skywork-OR1-RL-Data (He et al., 2025). We screen this pool with the base student Qwen3-4B, retaining challenging problems with low empirical success rates. After removing problems with invalid answers and removing overlap with out-of-distribution benchmarks, we sample 30k training, 1k validation, and 2k held-out in-distribution test problems (Math-ID) from the filtered pool. We also evaluate on the out-of-distribution benchmarks AIME 2024/2025 (Li et al., 2024) and MATH500 (Hendrycks et al., 2021).

Models and training. We use Qwen3-1.7B as the student for logic games and Qwen3-4B for mathematical reasoning, with Qwen3-4B-Thinking as the teacher in both settings (Yang et al., 2025). All experiments use verl (Sheng et al., 2025) with GRPO (Shao et al., 2024). By default, we sample six student-only and two teacher-assisted rollouts per prompt, all with thinking mode enabled. The maximum response length is 8,192 tokens for logic games and 16,384 tokens for mathematical reasoning; by default, the teacher generates at most $\ell_v = 4{,}096$ tokens per intervention.

Evaluation. We evaluate all models with thinking mode enabled, reporting avg@8/pass@8 for the logic games, Math-ID, and MATH500, and avg@64/pass@64 for AIME 2024/2025. Detailed descriptions of the baselines, datasets, and training hyperparameters are provided in Section A.

Intervention	Survo		Minesweeper		Kakurasu	
	avg@8	pass@8	avg@8	pass@8	avg@8	pass@8
Prefix ($\ell_u = 0$)	45.90	87.00	14.81	42.50	0.57	3.20
Middle ($\ell_u = 512$)	50.70 +4.80	**90.40** +3.40	**19.25** +4.44	**55.50** +13.00	3.52 +2.95	**16.60** +13.40
Middle ($\ell_u = 1024$)	49.88 +3.98	86.40 -0.60	17.94 +3.13	55.00 +12.50	**3.77** +3.20	15.20 +12.00
Middle ($\ell_u = 2048$)	**51.55** +5.65	87.20 +0.20	12.50 -2.31	43.50 +1.00	3.32 +2.75	14.00 +10.80

Table 1: Effect of the intervention position ℓ_u on hard logic games. $\ell_u = 0$ is prefix intervention.

4.2 Where Should the Teacher Intervene?

To study how intervention timing affects student learning, we compare prefix intervention ($\ell_u = 0$) with intervention after a short student attempt ($\ell_u \in \{512, 1024, 2048\}$), keeping the teacher-segment budget fixed. Tab. 1 shows that $\ell_u = 512$ middle intervention improves avg@8 and pass@8 over prefix intervention across all three games, with the largest gains on Minesweeper and Kakurasu. We include a brief case study to further examine why middle intervention helps in practice.

Case study: Understanding the advantage of middle intervention

Why does middle intervention help? We consider two possible reasons:

1. **Later tokens are more informative.** Early tokens may largely restate the problem or organize its conditions. Moving supervision later could thus make better use of the same distillation budget.
2. **Student-conditioned guidance is more learnable.** Conditioning teacher guidance on the student's partial solution may make it easier for the student to learn how to continue or recover.

We test these explanations by replacing the student-generated prefix with a teacher-generated one, leaving the rest of the rollout unchanged. Since the teacher may provide a stronger starting point, if middle intervention helps by shifting supervision to more informative tokens, this variant should match or surpass student-prefix intervention.

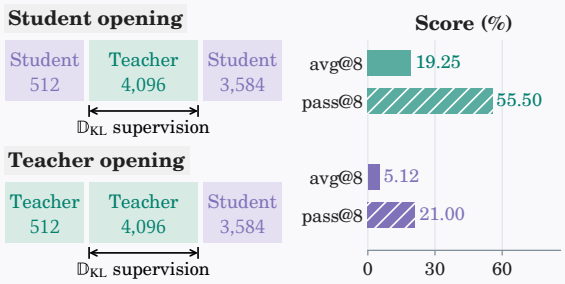

Figure 4: Effect of opening source on Minesweeper.

Yet this replacement substantially reduces performance (Fig. 4), supporting the second explanation. This result echoes the motivation behind DAgger (Ross et al., 2011) and on-policy distillation (Agarwal et al., 2024), which expose the learner to states from its own rollouts. From these states, the teacher can help the student build on partial reasoning or correct mistakes.

A longer student prefix, however, does not necessarily make this guidance more effective. On Minesweeper, for example, avg@8 falls from 19.25% at $\ell_u = 512$ to 12.50% at $\ell_u = 2048$, below the 14.81% achieved by prefix intervention. We also observe more truncation with later intervention. A longer student prefix leaves fewer tokens to continue generating after the teacher segment, especially when earlier steps need to be revised. As a result, the rollout may reach the length limit before the student can complete the revised reasoning and produce a final answer.

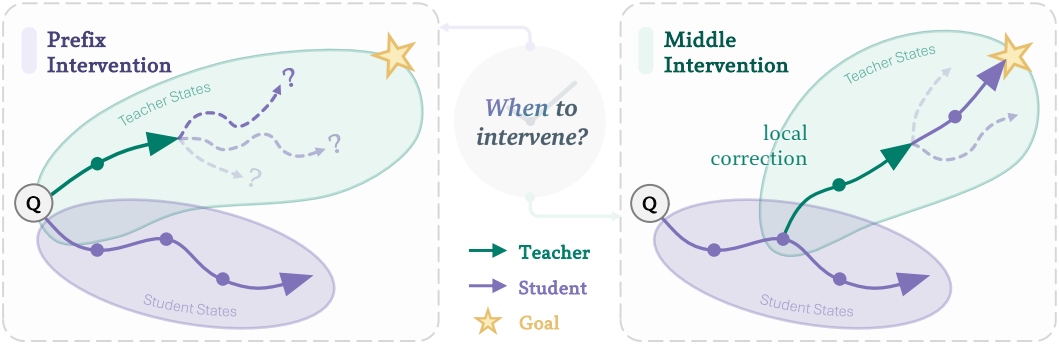

Figure 5: Prefix v.s. middle intervention. Guidance is more effective when the teacher can build on the student's partial solution and correct errors along the way.

Takeaway. Let the student make a short attempt, then guide it from its own reasoning. Intervene early enough to leave room for the student to revise its reasoning and complete the solution.

4.3 How Should the Teacher Supervise?

Beyond where the teacher intervenes, the form of supervision also matters for student learning. A common approach is to minimize the KL divergence between the student and teacher token distributions, providing dense token-level supervision (Agarwal et al., 2024; Xu et al., 2025). The forward- and reverse-KL objectives introduced in Section 2 both encourage distribution matching, but take expectations under the teacher and student token distributions, respectively. We compare these objectives within PIVOT to examine how the choice of KL direction affects learning from teacher-generated segments across tasks with different levels of initial competence.

Fig. 6 shows that reverse KL performs better on all three hard logic games, where the base student almost never succeeds: mean avg@8 rises from 24.5% to 35.6%, and mean pass@8 from 54.2% to 63.8%, relative to forward KL (Tab. 4). On math benchmarks, both objectives improve average accuracy over the base model. On the challenging Math-ID, where the base model achieves only 5.04% accuracy, reverse KL slightly outperforms forward KL by 0.57 percentage points. On the OOD benchmarks AIME 2024/2025 and MATH500, where the student already solves a substantial fraction of problems, forward KL yields modestly larger gains in average accuracy (Tab. 5).

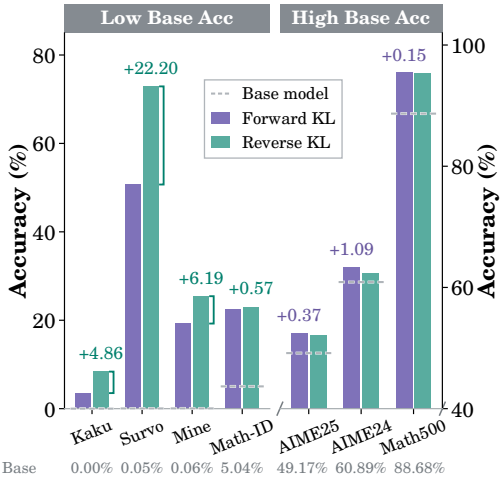

Figure 6: Reverse KL favors low initial accuracy; forward KL has a modest advantage at higher accuracy. Dashed lines mark base accuracy. Panels use different scales; labels show gains over the other KL, colored by the better KL.

A possible explanation lies in the mode-seeking and mode-covering tendencies of reverse and forward KL (Tajwar et al., 2024; Li et al., 2026). Reverse KL discourages the student from assigning probability to continuations the teacher considers unlikely, favoring concentration on teacher-preferred modes. When the student rarely succeeds, learning to reliably produce even one effective reasoning mode could yield substantial progress. With reliable teacher guidance, this concentration may help the student establish a solution strategy that it previously struggled to produce independently. Once successful reasoning is already accessible, learning additional effective strategies may offer further room for improvement across different problems. Forward KL encourages coverage of teacher-supported alternatives, including those the student currently assigns little probability, and may therefore support this broader learning. The results are consistent with this interpretation, though they do not directly measure changes in reasoning-mode coverage.

We therefore favor reverse KL, given its substantial gains on hard tasks and the modest accuracy advantage of forward KL when initial success rates are higher.

Takeaway. The student's starting point can inform the choice of KL objective. When success is rare, reverse KL's mode-seeking tendency may help the student establish a viable reasoning strategy. Forward KL offers a modest accuracy advantage at higher initial success rates, whereas reverse KL yields larger gains on hard tasks.

4.4 What Else Matters for Teacher Guidance?

With the intervention position and supervision objective fixed, we investigate which other aspects of teacher guidance affect student learning. We vary teacher capacity, the allocation of a fixed teacher-token budget across assisted rollouts, and the support size of the top-k approximation. Fig. 7 summarizes ablations, which vary one factor at a time while keeping other settings fixed.

Teacher model capacity. We examine whether the student benefits from a stronger teacher by replacing Qwen3-4B-Thinking with Qwen3-30B-A3B-Thinking. The stronger teacher improves avg@8 across all three games: performance more than doubles on Minesweeper, from 19.25% to

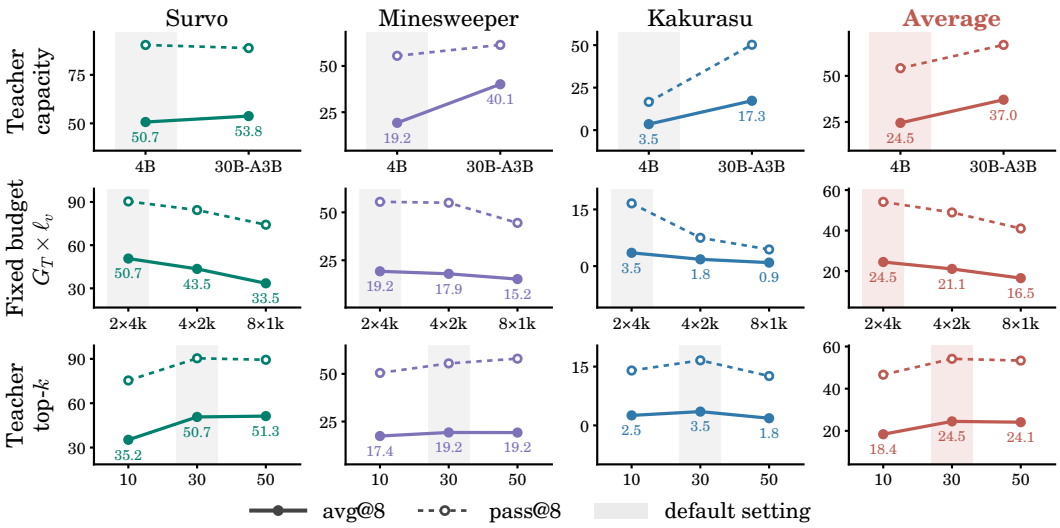

Figure 7: Calibrating teacher guidance on hard logic games under middle intervention with forward KL. Each column varies one factor from the default setting, which is shaded. Solid lines denote avg@8 and dashed lines denote pass@8. Full numbers are given in Tab. 6 in the appendix.

40.12%, and increases nearly fivefold on Kakurasu, from 3.52% to 17.27%, while the improvement on Survo is more modest. These results indicate that PIVOT can make effective use of a more capable teacher despite the larger teacher–student capacity gap. The additional capability of the teacher translates into improvements in the student's own problem-solving performance, even on games where learning with the default teacher remains difficult.

Allocation of a fixed teacher budget. We examine how a fixed teacher-token budget should be distributed across assisted rollouts. Keeping $G_T \times \ell_v$ fixed at 8k tokens, we allocate this budget to either two assisted rollouts with a 4k-token teacher budget each, four with 2k each, or eight with 1k each. Both avg@8 and pass@8 decline monotonically across all three games as the budget is spread across more rollouts; on Survo, avg@8 falls from 50.70% to 43.50% and 33.50%. This suggests that longer teacher segments give the student more opportunity to learn from a sustained sequence of connected reasoning or corrective steps, while shorter segments may end before the teacher has fully developed the reasoning needed for the student to continue. Under the same teacher-token budget, fewer, longer interventions therefore provide more effective guidance in our experiments.

Top-k approximation. The top-k approximation controls how much of the teacher's token distribution is available for supervision. Our results suggest that effective guidance requires a sufficiently broad set of teacher-supported alternatives, but does not necessarily benefit from retaining more of the distribution. Moving from $k = 10$ to $k = 30$ improves both metrics across all three games, whereas increasing k to 50 offers no consistent benefit and reduces performance on Kakurasu. This pattern suggests that the most probable teacher tokens alone may provide too narrow a learning signal, while extending supervision to lower-probability alternatives is not uniformly useful. We therefore use $k = 30$ as a balance between supervision quality and the cost of storing and matching teacher logits.

Takeaway. A stronger teacher provides better guidance, but making effective use of that guidance also requires careful allocation. With a fixed teacher-token budget, fewer, longer interventions are more effective than many short ones. For KL supervision, the teacher distribution should cover enough alternatives for learning; extending to lower-probability tokens offers no consistent benefit.

4.5 Putting the Recipe into Practice

Following the recipe above, we configure PIVOT to intervene after a 512-token student prefix, allowing the teacher to build on the student's initial reasoning. For each prompt, we sample six student-only and two teacher-assisted rollouts, concentrating the teacher-token budget on fewer,

Method	Hard Logic Games (Qwen3-1.7B)						Math Benchmarks (Qwen3-4B)							
	Survo		Minesweeper		Kakurasu		Math-ID		AIME 2024		AIME 2025		MATH500	
Base model	0.05		0.06		0.00		5.04		60.89		49.17		88.68	
Baselines														
RL only	0.05		1.94	+1.88	0.00		20.70	+15.66	60.62	-0.27	49.74	+0.57	94.30	+5.62
SFT+RL	39.77	+39.72	13.44	+13.38	1.38	+1.38	8.82	+3.78	60.94	+0.05	49.21	+0.04	89.98	+1.30
LUFFY	1.30	+1.25	0.81	+0.75	0.00		16.66	+11.62	60.68	-0.21	46.93	-2.24	90.60	+1.92
POPE	0.03	-0.02	0.00	-0.06	0.00		22.09	+17.05	60.00	-0.89	46.61	-2.56	94.20	+5.52
KDRL	41.68	+41.63	13.88	+13.82	2.93	+2.93	17.11	+12.07	49.11	-11.78	43.39	-5.78	92.92	+4.24
PIVOT														
Forward KL	50.70	+50.65	19.25	+19.19	3.52	+3.52	22.48	+17.44	**63.33**	+2.44	**52.45**	+3.28	**95.40**	+6.72
Reverse KL	**72.90**	+72.85	**25.44**	+25.38	**8.38**	+8.38	**23.05**	+18.01	62.24	+1.35	52.08	+2.91	95.25	+6.57

Table 2: Main results on hard logic games (Qwen3-1.7B) and math (Qwen3-4B). Math-ID is in-distribution; other math benchmarks are OOD. The base model uses thinking mode. Metrics are avg@8, or avg@64 for AIME. Colored boxes show absolute changes over the base model.

longer interventions. We apply teacher–student KL supervision to the teacher-generated segments with a top-k support of 30. We favor reverse KL for its substantial gains when initial success is rare, while also reporting the forward-KL variant. Tab. 2 compares these configurations with baselines.

On the hard logic games, the base student rarely produces a correct solution, and training with outcome rewards alone does little to improve its performance. Adding teacher-generated segments to rollouts, as in LUFFY and POPE, brings only limited improvements, with both methods performing worse than even the simple SFT+RL baseline. KDRL provides token-level supervision along student rollouts and makes greater progress, achieving the strongest baseline performance on all three games. PIVOT goes further by pairing teacher-generated reasoning segments with direct supervision on those segments: with reverse KL, it raises mean avg@8 from KDRL's 19.5% to 35.6%. A similar advantage appears on the hard math split, where the base model solves only 5.04% of problems. Here, PIVOT reaches 23.05%, exceeding POPE, the strongest baseline, at 22.09%.

On AIME 2024/2025 and MATH500, where the base student already solves a substantial fraction of problems, teacher guidance does not consistently improve performance. LUFFY, POPE, and KDRL improve on MATH500 yet fall below even the base model on both AIME benchmarks, while SFT+RL brings little improvement on either. PIVOT outperforms baselines across all three benchmarks with either KL objective, with forward KL achieving the highest accuracy. PIVOT therefore benefits students both when initial success is rare and when reasoning abilities are already present.

5 Conclusion

Effective teacher guidance in RLVR should support successful reasoning while remaining learnable for the student. In this work, we identify two complementary mechanisms for achieving this goal: *trajectory-level intervention* and *token-level supervision*. Existing methods typically emphasize one of them, but our analysis shows that either alone can be insufficient for student learning. Building on this insight, we introduce PIVOT, which integrates teacher corrections into student rollouts and applies token-level supervision to those corrections. We examine where to intervene, how to supervise, and how to allocate teacher guidance, yielding a recipe for student learning. Experiments on logic games and mathematical reasoning demonstrate the effectiveness of combining these mechanisms.

Limitations. Our study has several limitations. First, our evaluation focuses on logic games and mathematical reasoning; applicability to code generation, instruction following, and open-ended reasoning remains to be explored. Second, PIVOT uses fixed intervention and supervision settings throughout training, even as the student's capabilities evolve. How to effectively adapt teacher guidance to the student's changing needs remains an open question for future work.

AI USE STATEMENT

We used generative AI tools to assist with coding and with drafting, revising, and polishing the manuscript. The authors reviewed AI-assisted code and text, verified the scientific claims and citations, and remain responsible for the research design, failure analysis, and interpretation of results. No human-subject data were collected. We take full responsibility for the manuscript, reported results, and accompanying artifacts, including AI-assisted content.

REPRODUCIBILITY STATEMENT

We describe dataset construction, student and teacher models, training schedules, optimization settings, teacher-intervention budgets, KL-supervision settings, and evaluation protocols in Appendix A. Full experimental results are provided in Appendix B. After the review process concludes, we will open-source the complete training code and publicly release the PIVOT model checkpoints to support reproduction of our results and further research on teacher-guided reinforcement learning with verifiable rewards.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024.

Chenxin An, Zhihui Xie, Xiaonan Li, Lei Li, Jun Zhang, Shansan Gong, Ming Zhong, Jingjing Xu, Xipeng Qiu, Mingxuan Wang, et al. Polaris: A post-training recipe for scaling reinforcement learning on advanced reasoning models, 2025. *URL https://hkunlp. github. io/blog/2025/Polaris*, 2, 2025.

Justin Chih-Yao Chen, Becky Xiangyu Peng, Prafulla Kumar Choubey, Kung-Hsiang Huang, Jiaxin Zhang, Mohit Bansal, and Chien-Sheng Wu. Nudging the boundaries of llm reasoning. *arXiv preprint arXiv:2509.25666*, 2025.

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.

Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, et al. Skywork open reasoner 1 technical report. *arXiv preprint arXiv:2505.22312*, 2025.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Li Jiang, Haoran Xu, Yichuan Ding, and Amy Zhang. Trajectory-refined distillation. *arXiv preprint arXiv:2606.08432*, 2026.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, et al. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*, 13(9):9, 2024.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huanang 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.

Junteng Liu, Yuanxiang Fan, Jiang Zhuo, Han Ding, Yongyi Hu, Chi Zhang, Yiqi Shi, Shitong Weng, Aili Chen, Shiqi Chen, et al. Synlogic: Synthesizing verifiable reasoning data at scale for learning logical reasoning and beyond. *Advances in Neural Information Processing Systems*, 38: 100976–100997, 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.

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.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025.

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

Leitian Tao, Ilia Kulikov, Swarnadeep Saha, Tianlu Wang, Jing Xu, Sharon Li, Jason E Weston, and Ping Yu. Hybrid reinforcement: When reward is sparse, it's better to be dense. *arXiv preprint arXiv:2510.07242*, 2025.

Liang Wen, Yunke Cai, Fenrui Xiao, Xin He, Qi An, Zhenyu Duan, Yimin Du, Junchen Liu, Tanglifu Tanglifu, Xiaowei Lv, et al. Light-r1: Curriculum sft, dpo and rl for long cot from scratch and beyond. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pp. 318–327, 2025.

Yangzhen Wu, Shanda Li, Zixin Wen, Xin Zhou, Ameet Talwalkar, Yiming Yang, Wenhao Huang, and Tianle Cai. Learn hard problems during rl with reference guided fine-tuning. *arXiv preprint arXiv:2603.01223*, 2026.

Hongling Xu, Qi Zhu, Heyuan Deng, Jinpeng Li, Lu Hou, Yasheng Wang, Lifeng Shang, Ruifeng Xu, and Fei Mi. Kdrl: Post-training reasoning llms via unified knowledge distillation and reinforcement learning. *arXiv preprint arXiv:2506.02208*, 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. *arXiv preprint arXiv:2504.14945*, 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. *Advances in Neural Information Processing Systems*, 38:117157–117186, 2026.

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.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025.

Rosie Zhao, Alexandru Meterez, Sham Kakade, Cengiz Pehlevan, Samy Jelassi, and Eran Malach. Echo chamber: Rl post-training amplifies behaviors learned in pretraining. *arXiv preprint arXiv:2504.07912*, 2025.

A EXPERIMENTAL SETUP

We detail the dataset construction, training configuration, evaluation protocol, and baseline implementations.

Dataset construction. For synthetic logic games, we follow SynLogic (Liu et al., 2026) and construct three datasets: Survo, Minesweeper, and Kakurasu. Each game is generated at a hard difficulty setting, controlled by grid size, the number of unknown entries, and constraint density, at which the base student is near zero accuracy. Each game has 2k training problems and a held-out test split of 500 problems. For mathematical reasoning, we pool problems from Polaris-53K (An et al., 2025) and the math subset of Skywork-OR1-RL-Data (He et al., 2025), and screen them with the base student: Qwen3-4B samples 8 responses per problem, and we keep problems solved at most 3 times. After removing problems with invalid gold answers and decontaminating against AIME 2024/2025 and MATH500, we sample 30k problems for training, 1k for validation, and a held-out in-distribution test split of 2k problems, denoted Math-ID; 77.7% of the training problems are never solved by the base student.

Training setup. All experiments are conducted using the verl framework (Sheng et al., 2025). We use GRPO (Shao et al., 2024) as the policy optimization algorithm, with reference KL coefficient $\beta_{\text{ref}} = 0.01$ and clipping coefficient $\varepsilon = 0.2$. Unless otherwise specified, we use a learning rate of 3×10^{-6}, a rollout batch size of 64 prompts, a mini-batch size of 32, 8 rollouts per prompt, a maximum prompt length of 2,048 tokens, and sampling temperature 0.7 for both student and teacher. Logic-game experiments use Qwen3-1.7B as the student and Qwen3-4B-Thinking as the teacher (Yang et al., 2025); we train for 3 epochs with a maximum response length of 8,192 tokens and compute rewards with the SynLogic verifier. Math experiments use Qwen3-4B as the student and the same teacher; we train for 1 epoch with a maximum response length of 16,384 tokens and compute rewards with Math-Verify. No additional format or length rewards are used.

PIVOT configuration. For each prompt, we sample $G_S + G_T = 8$ rollouts, where $G_S = 6$ are student-only rollouts and $G_T = 2$ are teacher-assisted rollouts. By default, the teacher intervenes after a student prefix of $\ell_u = 512$ tokens and generates at most $\ell_v = 4{,}096$ tokens for logic games and $\ell_v = 8{,}192$ tokens for math. The student then resumes autoregressive generation from the teacher-augmented context. The teacher-generated segment is supervised with a token-level KL objective toward the teacher distribution; we use forward KL by default and study the reverse-KL variant in Section 4.3. The KL is computed on the teacher's top-30 tokens, with the remaining probability mass collected into a single tail bucket. The teacher KL coefficient β_{T} is cosine-decayed from 0.1 to 0. Ablations on the intervention position ℓ_u, the teacher model, the allocation of the teacher-token budget between G_T and ℓ_v, and the distillation top-k are provided in Section 4.4, with full numbers in Tables 1 and 6.

Evaluation. For synthetic logic games, we evaluate models on the held-out hard splits and report avg@8 and pass@8. For mathematical reasoning, we evaluate in distribution on Math-ID and out of distribution on MATH500 (Hendrycks et al., 2021) and AIME 2024/2025 (Li et al., 2024), reporting avg@8 and pass@8 for Math-ID and MATH500 and avg@64 and pass@64 for AIME 2024/2025. Decoding uses temperature 0.7 and the training response-length limit.

Baseline methods. We compare PIVOT with five representative baselines that address sparse rewards in different ways. All baselines share the GRPO setup above and the budget of 8 rollouts per prompt. RL only is the standard on-policy RLVR baseline without a teacher. SFT+RL first fine-tunes the student on 5k verified-correct teacher-generated traces from the same training distribution and then continues RLVR training. LUFFY (Yan et al., 2025) replaces one of the 8 rollouts with a full teacher-generated trajectory and trains on it with LUFFY's policy-shaping objective ($\gamma = 0.1$). POPE (Qu et al., 2026) starts 2 of the 8 rollouts from a teacher-generated prefix whose length matches our default segment length ℓ_v (4,096 tokens for logic games, 8,192 for math); prefix tokens are excluded from all losses and the student completes the rollout. KDRL (Xu et al., 2025) adds no teacher tokens to the trajectory; it combines GRPO with a reverse-KL distillation term on every student-generated token, computed on the student's top-30 tokens with a constant coefficient of 0.1.

Method	Survo				Minesweeper				Kakurasu			
	avg@8		pass@8		avg@8		pass@8		avg@8		pass@8	
Base model	0.05		0.40		0.06		0.50		0.00		0.00	
Baselines												
RL only	0.05		0.40		1.94	+1.88	8.00	+7.50	0.00		0.00	
SFT+RL	39.77	+39.72	86.40	+86.00	13.44	+13.38	41.50	+41.00	1.38	+1.38	7.60	+7.60
LUFFY	1.30	+1.25	9.00	+8.60	0.81	+0.75	5.00	+4.50	0.00		0.00	
POPE	0.03	-0.02	0.20	-0.20	0.00	-0.06	0.00	-0.50	0.00		0.00	
KDRL	41.68	+41.63	81.40	+81.00	13.88	+13.82	42.50	+42.00	2.93	+2.93	12.40	+12.40
PIVOT												
Forward KL	50.70	+50.65	90.40	+90.00	19.25	+19.19	55.50	+55.00	3.52	+3.52	16.60	+16.60
Reverse KL	**72.90**	+72.85	**97.40**	+97.00	**25.44**	+25.38	**63.00**	+62.50	**8.38**	+8.38	**31.00**	+31.00

Table 4: Full results on hard logic games (Qwen3-1.7B). We report avg@8 and pass@8; the base model uses thinking mode. Colored boxes show absolute changes over the base model.

B Additional Experimental Results

We report additional comparisons and full benchmark and ablation results. Colored boxes indicate changes relative to the corresponding base model unless otherwise specified.

B.1 Prefix-Only Guidance

Method	avg@8	pass@8
POPE w/ noisy prefix	0.03	0.20
POPE w/ oracle prefix	0.14	0.91
Prefix-Fwd	59.05	94.20

Table 3: Prefix-only guidance vs. prefix supervision on Survo. Oracle prefixes are verifier-filtered teacher rollouts.

Table 3 compares POPE-style prefix-only guidance with supervised prefix training on Survo. The oracle-prefix variant uses verifier-filtered teacher rollouts as privileged prefixes. Although oracle prefixes are stronger than noisy teacher prefixes, both variants remain close to zero accuracy, suggesting that using teacher prefixes only as context provides limited learning signal.

B.2 Hard Logic Games

Table 4 reports the full avg@8 and pass@8 results on hard logic-game test splits.

B.3 Mathematical Reasoning

Table 5 reports the full math results, including the held-out in-distribution test split (Math-ID). We report avg@8 and pass@8 for Math-ID and MATH500, and avg@64 and pass@64 for AIME 2024/2025.

B.4 Teacher-Guidance Ablations

Table 6 lists the complete numbers behind Figure 7; each block varies one factor from the default configuration, which is repeated in every block for reference. The prefix-length ablation is reported in Table 1.

Method	Math-ID				AIME 2024				AIME 2025				MATH500			
	avg@8		pass@8		avg@64		pass@64		avg@64		pass@64		avg@8		pass@8	
Base model	5.04		22.20		60.89		83.33		49.17		76.67		88.68		92.20	
Baselines																
RL only	20.70	+15.66	37.00	+14.80	60.62	-0.27	83.33		49.74	+0.57	76.67		94.30	+5.62	98.00	+5.80
SFT+RL	8.82	+3.78	28.20	+6.00	60.94	+0.05	83.33		49.21	+0.04	76.67		89.98	+1.30	94.20	+2.00
LUFFY	16.66	+11.62	33.00	+10.80	60.68	-0.21	83.33		46.93	-2.24	76.67		90.60	+1.92	97.20	+5.00
POPE	22.09	+17.05	38.90	+16.70	60.00	-0.89	83.33		46.61	-2.56	76.67		94.20	+5.52	97.60	+5.40
KDRL	17.11	+12.07	30.20	+8.00	49.11	-11.78	76.67	-6.66	43.39	-5.78	63.33	-13.34	92.92	+4.24	96.80	+4.60
PIVOT																
Forward KL	22.48	+17.44	41.70	+19.50	**63.33**	+2.44	**86.67**	+3.34	**52.45**	+3.28	**80.00**	+3.33	**95.40**	+6.72	**98.80**	+6.60
Reverse KL	**23.05**	+18.01	**42.75**	+20.55	62.24	+1.35	83.33		52.08	+2.91	76.67		95.25	+6.57	98.60	+6.40

Table 5: Full math results (Qwen3-4B, thinking mode). We report avg@8/pass@8 for Math-ID and MATH500, and avg@64/pass@64 for AIME. Colored boxes show absolute changes over the base model.

Setting	Survo		Minesweeper		Kakurasu	
	avg@8	pass@8	avg@8	pass@8	avg@8	pass@8
Teacher capacity						
Qwen3-4B-Thinking ★	50.70	**90.40**	19.25	55.50	3.52	16.60
Qwen3-30B-A3B-Thinking	**53.75**	88.80	**40.12**	**61.40**	**17.27**	**50.20**
Fixed teacher budget ($G_T \times \ell_v = 8$k)						
2×4k ★	**50.70**	**90.40**	**19.25**	**55.50**	**3.52**	**16.60**
4×2k	43.50	84.40	17.94	55.00	1.80	7.50
8×1k	33.50	74.20	15.19	44.50	0.93	4.40
Teacher top-k						
$k = 10$	35.23	75.40	17.38	50.50	2.55	14.00
$k = 30$ ★	50.70	**90.40**	**19.25**	55.50	**3.52**	**16.60**
$k = 50$	**51.27**	89.40	19.19	**58.00**	1.83	12.60

Table 6: Teacher-guidance ablations with middle intervention and forward KL. Each block varies one factor; shaded rows marked with ★ repeat the default setting. Bold: best within each block.

C Case Studies on Intervention Placement

Figures 8 and 9 trace one Survo puzzle through the three intervention positions of Table 1 (trained student, step 60). For each $\ell_u \in \{0, 512, 2048\}$ we take the teacher-assisted rollout of this puzzle from the corresponding training run and freeze the context up to the end of the teacher segment: a ℓ_u-token student prefix followed by a 4,096-token teacher segment. Under the 8,192-token response limit this leaves 4,096, 3,584, and 2,048 tokens for the continuation. Each frozen context is continued four times (temperature 0.7, same remaining budget) by the teacher and by one fixed student (the $\ell_u = 512$ checkpoint at step 60), so the two completion rates are measured from exactly the same state. The left column of each box is the skeleton of the trajectory in generation order, with verbatim excerpts and token offsets (`[+k]` counts from the start of the continuation); each numbered excerpt is explained by the annotation with the same number on the right. Pink marks wasted or failing reasoning, green marks progress.

Case 1: prefix ($\ell_u = 0$) vs. middle ($\ell_u = 512$) intervention on the same Survo puzzle

TRAJECTORY SKELETON (GENERATION ORDER)	CONCLUSION — EXPLANATION
Prompt (shared by all placements) 8 2 7 8 X 2 32 5 5 6 5 7 1 29 3 6 7 X 5 9 32 2 6 8 X X 4 31 7 3 X 5 X 6 35 X X 5 9 X X 38 29 30 39 35 34 30 197 Fill the 10 cells marked X with [5, 8, 4, 6, 2, 6, 4, 8, 8, 5], each used once, so that the last number of every row and column is the sum of the others. Answer with the filled matrix in a Python code block.	Same prompt and 8,192-token limit; only intervention timing differs.
Prefix intervention ($\ell_u = 0$)	
Teacher segment, tokens 0–4,096	
❶ [0.5k] *I think I have a misunderstanding. Let me read the problem again.* [1.0k] *I think there's confusion in indexing.*	❶ **Slow start.** Re-reads the task for ∼1.5k tokens.
❷ [2.5k] *Yes, so row 0, col4 must be 5.* [3.0k] *So row 2, col3 must be 2.*	❷ **Little progress.** Only 2/10 cells solved by hand-off.
❸ [3.4k] *Let's use different notation. [. . .] let A = M[3][3], B = M[3][4], so A + B = 11 [. . .] E + F + G + H = 24* [4.1k] (segment ends)	❸ **Teacher notation.** Eight open cells are renamed $A, \ldots, H$; the student must continue within this inherited system.
Student continuation, budget 4,096	
❹ [+0.5k] *with G + H = 24 [. . .] 14 < 24. Contradiction. Wait, this can't be. What's wrong?* [+2.4k] *Contradiction again.* [+4.1k] (length limit, no answer)	❹ **Student 0/4.** Drops E, F and loops over contradictions, or solves the grid but keeps re-checking until truncation.
❺ **Teacher from the same hand-off** [+0k] *thus E = 4 [. . .] thus C = 6 [. . .] All good.* [+3.7k] ```python	❺ **Teacher 4/4.** Completes from the same hand-off: the state is solvable, but difficult for the student to finish.
Middle intervention ($\ell_u = 512$)	
Student prefix, tokens 0–512	
❶ [0.1k] *Row 0: [8, 2, 7, 8, 'X', 2, 32] [. . .] Row 6: [29, 30, 39, 35, 34, 30, 197]* [0.5k] *Let me count.*	❶ **Student's set-up.** Lists rows and states the sum rule, giving the teacher a context built by the student.
Teacher segment, tokens 512–4,608	
❷ [0.6k] *8+2+7+8+2 = 27, so 27 + X = 32, so X = 5.*	❷ **Immediate progress.** First cell 87 tokens after take-over, versus 2.5k from scratch; the teacher uses the student's notation.
❸ [2.9k] *E is (5,0), so (5,0) = 4* [3.3k] *thus C = 6* [4.1k] *thus H=8* [4.3k] *Perfect.* [4.6k] (segment ends)	❸ **Solved at hand-off.** All 10 cells are fixed and verified, leaving 3,584 tokens for the student to finish the response.
Student continuation, budget 3,584	
❹ [+1.4k] *All good.* </think> ```python	❹ **Student and teacher 4/4.** The student finishes in 1.4k–1.8k tokens, successfully continuing from its own teacher-extended reasoning.

Reading. Both states are solvable for the teacher, but only the student-conditioned middle state is completed by the student (4/4 vs. 0/4).

Figure 8: Case study 1 on one Survo puzzle (trained student, step 60): the state produced by prefix intervention is solvable for the teacher but not for the student, whereas the middle state is finished by both.

Case 2: a late intervention ($\ell_u = 2048$) leaves too little budget to finish (same puzzle)

TRAJECTORY SKELETON (GENERATION ORDER)	CONCLUSION — EXPLANATION
Student prefix, tokens 0–2,048	
❶ [0.5k] *So there are 12 empty cells, but the list says 12? [. . .] total 1+2+2+2+2=9? [. . .] = 10 empty cells.* [1.8k] *Let me write the equations.*	❶ **No progress.** Recounts cells; 0/10 solved after 2,048 tokens.
Teacher segment, tokens 2,048–6,144	
❷ [2.3k] *thus X=5.* [2.4k] *So row2, col3 should be 2.* [6.1k] *Let I = (4,2), J = (4,4) = D. So row4: col0=7, col1=3, col2=I, col3=* (segment cap)	❷ **Partial recovery.** Solves 6/10 cells before the cap; 2,048 tokens remain.
Student continuation, budget 2,048	
❸ [+0.6k] *D=8, so (4,4)=8 [. . .] So (4,2)=6*	❸ **Grid solved.** Last four cells fixed within 0.7k tokens.
❹ [+2.0k] *All good. Now, write the matrix in Python* (length limit)	❹ **Student and teacher 0/4.** Re-checking exhausts the budget before output.
❺ **Teacher with an 8,192-token budget** [+2.0k] ```python [+2.3k] </think>	❺ **More budget: teacher 4/4.** Needs 2.4k–3.8k tokens to finish.

Reading. Late intervention leaves too little budget to emit the answer; extra budget restores teacher success.

Figure 9: Case study 2 on the same Survo puzzle: with $\ell_u = 2048$, teacher and student both solve the remaining cells from the hand-off state but run out of budget before emitting the answer.