AIPO: LEARNING TO REASON FROM ACTIVE INTERACTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advances in large language models (LLMs) have demonstrated strong reasoning capabilities, largely stimulated by reinforcement learning with verifiable rewards (RLVR). However, the exploration of existing RLVR algorithms remains largely constrained by the knowledge and reasoning strategies already accessible to the policy model. Although recent methods introduce external expert demonstrations to broaden exploration, they typically rely on complete trajectory-level guidance, which can be sample-inefficient, information-sparse, and insufficiently adaptive to intermediate reasoning bottlenecks. Inspired by collaborative multi-agent systems, we propose **AIPO**, an enhanced reinforcement learning framework that improves LLM reasoning through active multi-agent interaction during exploration. Specifically, when encountering reasoning bottlenecks, AIPO enables the policy model to proactively consult three functional collaborative agents, namely the *verify agent*, *knowledge agent*, and *reasoning agent*, thereby obtaining fine-grained and state-dependent guidance during rollout. The resulting mixed-policy trajectories expose the policy to reasoning directions that may be difficult to discover through isolated on-policy exploration. To learn effectively from collaborator-provided tokens, we further introduce a corrected importance sampling coefficient together with a lower-bound clipping strategy to mitigate off-policy discrepancy and vanishing gradients. After training, the policy model reasons independently without relying on collaborative agents. Extensive experiments across mathematical, scientific, coding, and puzzle reasoning benchmarks show that AIPO consistently improves reasoning performance and generalizes across different policy models, collaborator backbones, and RLVR algorithms. Additional out-of-distribution and training-dynamics analyses indicate that active interaction broadens the policy model's *effective exploration boundary*, enabling it to solve additional problems independently under the same inference-time sampling budget.

1 INTRODUCTION

Large language models (LLMs) have demonstrated strong reasoning abilities, enabling them to solve complex mathematical, coding, and scientific tasks (OpenAI, 2024a;b; 2025; DeepSeek-AI et al., 2025; Kimi-Team et al., 2025; Yang et al., 2025; Comanici et al., 2025; Huang & Yang, 2025; xAI, 2025). A key factor behind this progress is Chain-of-Thought (CoT) reasoning, where models explore and reflect over intermediate reasoning steps to construct robust reasoning processes before producing final answers (Chen et al., 2025c). These capabilities are largely stimulated by reinforcement learning with verifiable rewards (RLVR) (Schulman et al., 2017; Shao et al., 2024), which encourages LLMs, as policy models, to explore diverse reasoning trajectories and learn from the successful trajectories under verifiable reward signals, thereby acquiring stronger reasoning policies that support self-improvement and scalable inference-time reasoning (Snell et al., 2024).

Despite these advances, existing RLVR frameworks remain constrained by the knowledge and reasoning strategies that the policy model can access during its own rollouts, as illustrated in Figure 1 (A) (Yue et al., 2025a; Liao et al., 2025b; Chen et al., 2025a; He et al., 2025a; Zhang et al., 2025c; Fu et al., 2025). Although RLVR encourages exploration over diverse reasoning trajectories, the reachable trajectory space is still largely determined by the policy's pretrained knowledge and current reasoning behavior. Consequently, standard on-policy RLVR can improve performance by exploring more effectively within this accessible region, but may still fail to expose the policy to useful reasoning

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.

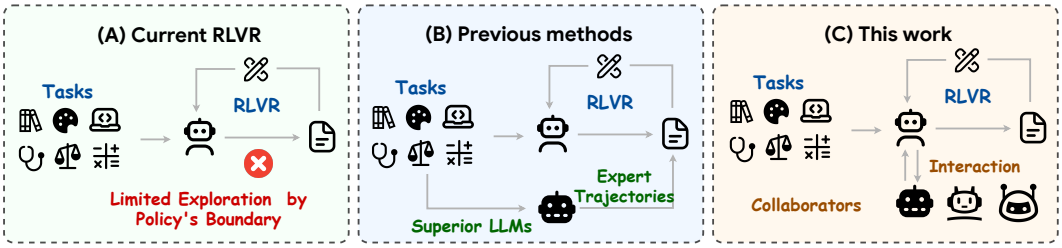

Figure 1: Comparison between existing methods and the proposed AIPO. (A) Standard on-policy RLVR explores trajectories generated by the policy model itself and is therefore constrained by its current exploration distribution. (B) Existing expert-guided approaches introduce supervision from externally generated expert trajectories, for example through supervised, distillation, or off-policy learning objectives. (C) AIPO instead introduces targeted collaborator feedback *during* rollout, producing mixed-policy trajectories conditioned on the policy's current reasoning bottlenecks.

directions that it is unlikely to discover independently (Yue et al., 2025a; Dang et al., 2025; Zhao et al., 2025).

To overcome this limitation, recent studies seek to enhance model performance and expand capability boundaries by leveraging guidance from stronger external expert models such as expert trajectories (DeepSeek-AI et al., 2025) or critiques (Lin et al., 2025), primarily through supervised fine-tuning (Chen et al., 2025b; Zhao et al., 2026) or offline reinforcement learning (Zhang et al., 2025b;c) with expert demonstrations, as illustrated in Figure 1 (B) (Lv et al., 2025). However, these methods typically depend on complete expert trajectories, which are costly to sample, information-sparse, and often redundant for training. Moreover, full-trajectory learning provides only coarse-grained supervision, offering limited fine-grained guidance for identifying and resolving intermediate reasoning bottlenecks (Deng et al., 2025). It also confines exploration to a static expert-generated distribution, leaving the trained model susceptible to the capability limits and biases of the static external expert.

Motivated by self-evolving and collaborative multi-agent systems (Xue et al., 2025; Liu et al., 2026; Chen et al., 2025d; Fu et al., 2025), which extend the capabilities of monolithic models through collaboration and communication among multiple LLMs, we propose **AIPO** (**A**ctive and **I**nteractive **P**olicy **O**ptimization). As illustrated in Figure 1 (C), AIPO enhances RLVR exploration through active interaction with external collaborative agents. Compared with methods based on complete expert demonstrations or trajectory-level critiques, AIPO provides fine-grained, state-dependent guidance at intermediate reasoning bottlenecks and does not necessarily require a stronger expert model. To provide complementary forms of assistance, AIPO introduces three functional collaborators: ❶ *Verify agent*, which verifies intermediate conclusions; ❷ *Knowledge agent*, which provides relevant domain knowledge; and ❸ *Reasoning agent*, which assists with intermediate sub-problems. During exploration, the policy autonomously decides whether and which collaborator to invoke, integrates the returned feedback into its reasoning, and thereby constructs *mixed-policy* trajectories that may contain reasoning directions unlikely to be discovered through isolated on-policy rollout. Then, the policy optimizes successful mixed-policy trajectories and internalizes their useful reasoning signals. Because these trajectories contain externally generated tokens, we introduce a corrected importance sampling coefficient and a lower-bound clipping strategy to mitigate the resulting off-policy discrepancy and vanishing-gradient problem. After training, all collaborators are removed and the policy model reasons independently using the knowledge and reasoning strategies internalized through interaction.

We conduct extensive experiments on diverse reasoning benchmarks, including AIME24, AIME25, MATH500 (Hendrycks et al., 2021), LiveMathBench (Liu et al., 2024), GPQA-Diamond (Rein et al., 2023), MBPP (Austin et al., 2021), LiveCodeBench (Jain et al., 2025), and Reasoning-Gym (Stojanovski et al., 2025). The results show that AIPO consistently improves over competitive RLVR, distillation, and expert-guided baselines across both in-domain and out-of-domain benchmarks. The gains generalize across different policy-model and collaborator backbones from the Qwen (Yang et al., 2024) and Llama (Dubey et al., 2024) families, and AIPO can be integrated with different RLVR algorithms, including GRPO (Shao et al., 2024), DAPO (Yu et al., 2025), and GSPO (Zheng et al., 2025). Further analyses on DeepMath and the training dynamics show that active interaction broadens the policy model's *effective exploration boundary*: after training, the policy solves additional problems independently without requiring collaborators or additional inference-time computation.

2 PRELIMINARIES

RLVR for LLM Reasoning. Reinforcement Learning with Verifiable Rewards (RLVR) is widely adopted to improve LLM reasoning (DeepSeek-AI et al., 2025; Kimi-Team et al., 2025; Yang et al., 2025; OpenAI, 2024a;b; Comanici et al., 2025; Hu et al., 2025). To avoid the separate value network required by PPO (Schulman et al., 2017; Ouyang et al., 2022), critic-free methods such as ReMax (Li et al., 2024), RLOO (Ahmadian et al., 2024), GRPO (DeepSeek-AI et al., 2025; Shao et al., 2024), and REINFORCE++ (Hu, 2025) replace the value estimate with reward-based baselines and optimize

$$\mathcal{J}(\theta)=\mathbb{E}_{q\sim P,\{\boldsymbol{\tau}_i\}\sim\pi_{\theta_{\text{old}}}}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|\boldsymbol{\tau}_i|}\sum_{t=1}^{|\boldsymbol{\tau}_i|}\left\{\min\left(\rho_{i,t}\tilde{A}_t^i,\ \text{clip}(\rho_{i,t},1-\epsilon,1+\epsilon)\,\tilde{A}_t^i\right)-\beta\, D_{\text{KL}}\left[\pi_\theta\|\pi_{\text{ref}}\right]\right\}\right], \tag{1}$$

where $\{\boldsymbol{\tau}_1,\ldots,\boldsymbol{\tau}_G\}\sim\pi_{\theta_{\text{old}}}(\cdot|q)$ is a group of G trajectories rolled out for a query $q\sim P$, $\tilde{A}_t^i$ is the normalized advantage computed from a reward baseline, $\rho_{i,t}=\frac{\pi_\theta(\boldsymbol{\tau}_{i,t}|\boldsymbol{\tau}_{i,<t})}{\pi_{\theta_{\text{old}}}(\boldsymbol{\tau}_{i,t}|\boldsymbol{\tau}_{i,<t})}$ is the importance sampling coefficient correcting the shift between the current and rollout policies, clipping bounds the update, and D_{KL} regularizes π_θ toward a reference policy π_{ref}.

Off-Policy Enhanced RLVR. To broaden the reasoning trajectories available to the policy beyond those produced by its own rollout distribution, recent methods (Zhang et al., 2025c; Lv et al., 2025; Zhang et al., 2025b; Chen et al., 2025b) introduce demonstrations from a stronger teacher π_{ext}, combining on-policy trajectories $\{\boldsymbol{\tau}_i\}_{i=1}^N\sim\pi_\theta(\cdot|q)$ with externally generated trajectories $\{\boldsymbol{\tau}_i'\}_{i=1}^M\sim\pi_{\text{ext}}(\cdot|q)$:

$$\mathcal{J}_c(\theta)=\mathcal{J}\left(\theta,\{\boldsymbol{\tau}_i\}_{i=1}^N\right)+\mathcal{J}_{\text{ext}}\left(\theta,\phi,\{\boldsymbol{\tau}_i'\}_{i=1}^M\right), \tag{2}$$

where ϕ denotes the teacher parameters and the auxiliary term $\mathcal{J}_{\text{ext}}$ is instantiated either as an SFT objective (Zhang et al., 2025b; Ma et al., 2025; Fu et al., 2025) or as an RLVR objective corrected for the *off-policy* discrepancy (Zhang et al., 2025c).

3 METHODOLOGY

In this section, we introduce AIPO, which consists of two complementary components: ❶ *exploration*, an active and interactive rollout process in which the policy selectively queries external collaborators to broaden the set of reachable reasoning trajectories as shown in Figure 2 (a) and § 3.1; and ❷ *learning*, in which the policy optimizes successful mixed-policy trajectories and internalizes the useful reasoning signals they contain (§ 3.2). Because mixed-policy trajectories additionally contain tokens generated by external collaborators, learning from them requires addressing the resulting off-policy discrepancy and vanishing-gradient problem (Figure 2 (b)).

3.1 ENHANCED ROLLOUT BASED ON ACTIVE INTERACTION

To broaden the policy model's effective exploration space, we design an enhanced rollout process based on active multi-agent interaction. Instead of requiring the policy to explore solely through trajectories generated from its own current distribution, the proposed rollout allows it to request targeted assistance when encountering intermediate reasoning bottlenecks. The resulting feedback introduces potentially useful reasoning directions into the rollout while preserving the policy's responsibility for integrating them into the final solution.

Overview of Active Interaction. Given a question q, the policy model π_θ first performs basic reasoning using its internal knowledge and capabilities, such as problem decomposition, solution planning, and elementary arithmetic operations (Wei et al., 2022). If the policy model can solve the question independently, external guidance is unnecessary, which is often overlooked by existing methods (Qin et al., 2024; Huang et al., 2024; Muennighoff et al., 2025; Guha et al., 2025). Conversely, when the policy model identifies uncertainty or an unresolved sub-problem during rollout, it may actively invoke an external collaborator to obtain targeted assistance. Under outcome-guided RLVR optimization, the policy can learn when and how to invoke collaborators, as interactions appearing in successful trajectories receive positive learning signals. Successful interactions can introduce useful reasoning signals into rewarded trajectories, which are subsequently reinforced through policy optimization.

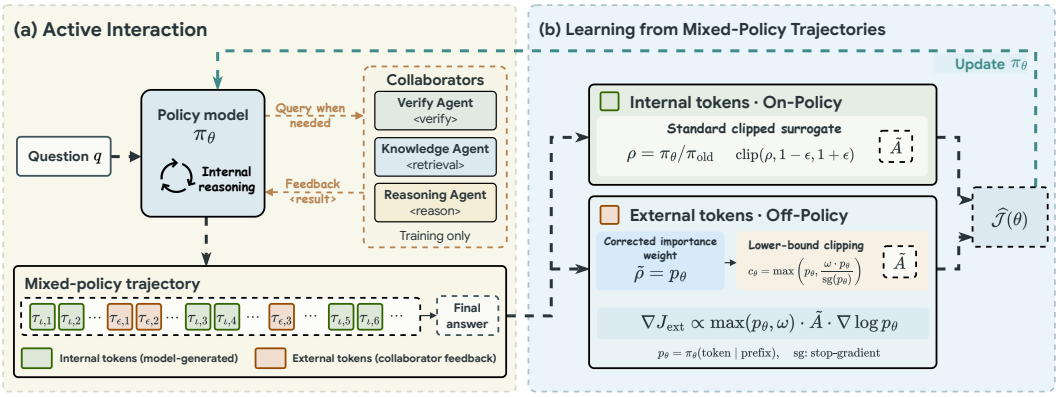

Figure 2: Illustration of AIPO. In the AIPO framework, during each rollout, the policy model engages in active interactions with collaborators. We then compute the reward and optimize the policy model using losses derived from both internal (on-policy) and external (off-policy) tokens. Additionally, we propose a corrected importance sampling coefficient and clipping strategy to mitigate off-policy issues and the vanishing gradient problem for external tokens.

Workflow of Active Interaction. Specifically, we provide the policy model with the definitions and descriptions of three functional agents, i.e., collaborators, in the system prompt (see Prompt B.1). These agents provide complementary forms of assistance for complex reasoning, and the policy interacts with them through a structured protocol. Detailed prompts are provided in § B.3.

- *Verify Agent:* Verifies the correctness of intermediate conclusions. The policy model invokes this agent by enclosing the conclusion to be checked within `<verify>` and `</verify>` tags, and the agent returns the verification result within `<result>` and `</result>` tags.
- *Knowledge Agent:* Provides knowledge required for reasoning. The policy model invokes this agent by enclosing its query within `<retrieval>` and `</retrieval>` tags, and the agent returns the retrieved information within `<result>` and `</result>` tags. Following ZeroSearch (Sun et al., 2025a), the Knowledge Agent synthesizes six short documents, including one noisy document, using the reference answer from the training data as hidden supervision. The policy observes only the generated documents, and the mechanism is used only during training. Further details are provided in § B.9.
- *Reasoning Agent:* Solves intermediate sub-tasks encountered during reasoning. The policy model invokes this agent by enclosing the sub-task within `<reason>` and `</reason>` tags, and the agent returns the corresponding result within `<result>` and `</result>` tags.

The responses from these agents are incorporated into the rollout trajectory as external tokens, producing mixed-policy trajectories that combine the policy model's own reasoning with collaborator-provided responses:

$$\boldsymbol{\tau} = \left\{\boldsymbol{\tau}_{\iota,1}, \ldots, \boldsymbol{\tau}_{\epsilon,1}, \ldots, \boldsymbol{\tau}_{\iota,t}, \ldots, \boldsymbol{\tau}_{\epsilon,|\boldsymbol{\tau}_\epsilon|}, \ldots, \boldsymbol{\tau}_{\iota,|\boldsymbol{\tau}_\iota|}\right\} \sim \prod_{i=1}^{|\boldsymbol{\tau}_\iota|} \pi_\theta\left(\boldsymbol{\tau}_{\iota,i} \mid \boldsymbol{\tau}_{<i}\right) \cdot \prod_{j=1}^{|\boldsymbol{\tau}_\epsilon|} \pi_\epsilon\left(\boldsymbol{\tau}_{\epsilon,j} \mid \boldsymbol{I}_j\right), \quad (3)$$

where $\boldsymbol{\tau}_\iota$ denotes the sequence of tokens generated by the policy model, $\boldsymbol{\tau}_\epsilon$ denotes the sequence of tokens generated by external collaborators, the subscript indexes each token, and $\boldsymbol{I}_j$ denotes the interaction context used to generate the j-th external token. For simplicity, we merge the three collaborator policies into a unified external distribution π_ϵ, while π_θ denotes the policy model. We provide additional discussions in § D.2.

Importantly, the policy model does not rely on external agents during inference. Instead, it reasons independently by leveraging the knowledge and strategies internalized during training. This distinguishes our method from conventional multi-agent systems (Wan et al., 2025; Gao et al., 2025; Liao et al., 2025a): interaction is used only to enhance training, whereas the trained model performs reasoning autonomously at test time.

3.2 Learning with Active Interaction

During interactive rollout, the policy model generates a trajectory $\boldsymbol{\tau}$ that consists of both internal tokens $\boldsymbol{\tau}_\iota$ generated by the policy model and external tokens $\boldsymbol{\tau}_\epsilon$ provided by collaborators. Unlike

previous methods (Jin et al., 2025; Song et al., 2025; Liu et al., 2025a), which typically discard external tokens and optimize only over the internal tokens generated by the policy model, AIPO incorporates both internal and external tokens into the learning process.

Off-Policy Issues. The primary challenge lies in learning from tokens generated by external collaborators. Standard RL objectives generally assume that all tokens are sampled from the current or previous policy distribution. The classical clipped surrogate objective is defined in Equation (1), where the importance sampling coefficient for each token $\boldsymbol{\tau}_{i,t}$ is $\rho_{i,t} = \frac{\pi_\theta(\boldsymbol{\tau}_{i,t}|\boldsymbol{\tau}_{i,<t})}{\pi_{\theta_{\text{old}}}(\boldsymbol{\tau}_{i,t}|\boldsymbol{\tau}_{i,<t})}$, and $\pi_{\theta_{\text{old}}}$ denotes the policy from the previous iteration. Directly applying Equation (1) to external tokens $\boldsymbol{\tau}_\epsilon$ may introduce off-policy issues due to distributional mismatch between the policy model and external collaborators, potentially destabilizing training (Schulman et al., 2017; Zhang et al., 2025c). Conventional approaches often avoid this issue by excluding external tokens from the policy loss (Song et al., 2025; Liu et al., 2025a). In contrast, AIPO explicitly incorporates external tokens into policy optimization, enabling the policy model to acquire useful knowledge and reasoning patterns from collaborators.

Correcting the Importance Sampling Coefficient for External Tokens. To mitigate off-policy errors, we introduce a corrected importance sampling coefficient $\tilde{\rho}$ for external tokens. Since external tokens are sampled from the collaborator distribution π_ϵ rather than $\pi_{\theta_{\text{old}}}$, the objective can be decomposed as

$$\mathcal{J}(\theta) = \mathbb{E}_{\pi_{\theta_{\text{old}}}}\left[\frac{1}{|\boldsymbol{\tau}_\iota|}\sum_{\boldsymbol{\tau}_t\in\boldsymbol{\tau}_\iota}\frac{\pi_\theta(\boldsymbol{\tau}_t \mid \boldsymbol{\tau}_{<t})}{\pi_{\theta_{\text{old}}}(\boldsymbol{\tau}_t \mid \boldsymbol{\tau}_{<t})}\tilde{A}_t\right] + \mathbb{E}_{\pi_\epsilon}\left[\frac{1}{|\boldsymbol{\tau}_\epsilon|}\sum_{\boldsymbol{\tau}_t\in\boldsymbol{\tau}_\epsilon}\frac{\pi_\theta(\boldsymbol{\tau}_t \mid \boldsymbol{\tau}_{<t})}{\pi_\epsilon(\boldsymbol{\tau}_t \mid \boldsymbol{I}_t)}\tilde{A}_t\right], \tag{4}$$

where clipping and KL-penalty terms are omitted for brevity. Although directly using the collaborator distribution in the denominator is theoretically natural, it poses two practical challenges: ❶ vocabulary discrepancies between the collaborator and the policy model may lead to incompatible token-level probability estimates; and ❷ evaluating and storing collaborator token distributions introduces additional computational and memory overhead. To address these limitations, we approximate the collaborator distribution as a one-hot distribution over the sampled external token and place it under the policy distribution, yielding the corrected objective:

$$\mathcal{J}'(\theta) = \mathbb{E}_{\pi_{\theta_{\text{old}}}}\left[\frac{1}{|\boldsymbol{\tau}_\iota|}\sum_{\boldsymbol{\tau}_t\in\boldsymbol{\tau}_\iota}\frac{\pi_\theta(\boldsymbol{\tau}_t \mid \boldsymbol{\tau}_{<t})}{\pi_{\theta_{\text{old}}}(\boldsymbol{\tau}_t \mid \boldsymbol{\tau}_{<t})}\tilde{A}_t\right] + \mathbb{E}_{\pi_{\theta_{\text{old}}}}\left[\frac{1}{|\boldsymbol{\tau}_\epsilon|}\sum_{\boldsymbol{\tau}_t\in\boldsymbol{\tau}_\epsilon}\pi_\theta(\boldsymbol{\tau}_t \mid \boldsymbol{\tau}_{<t})\tilde{A}_t\right]. \tag{5}$$

This approximation can be interpreted as assigning all probability mass to the observed collaborator token. It obviates explicit access to collaborator logits and their associated computation, resolves token-level vocabulary incompatibility, and still encourages the policy model to raise the likelihood of high-advantage external tokens. We provide further approximation error analysis in § A. Importantly, eliminating the explicit collaborator distribution from the coefficient relaxes a *distributional compatibility* requirement rather than a *quality* requirement. The collaborator need not share the policy's vocabulary, backbone, or token-level distribution, but its feedback must still contain useful reasoning signals for successful trajectories.

Gradient Vanishing for External Tokens. The standard surrogate objective uses clipping to prevent excessive policy deviation from the previous policy. For external tokens, however, the corrected coefficient $\pi_\theta(\boldsymbol{\tau}_t \mid \boldsymbol{\tau}_{<t})$ is naturally bounded by the softmax output. Thus, applying the original clipping mechanism is unnecessary. Without additional treatment, the gradient contribution of an external token is proportional to $\pi_\theta(\boldsymbol{\tau}_t \mid \boldsymbol{\tau}_{<t}) \cdot \tilde{A}_t \cdot \nabla_\theta \log \pi_\theta(\boldsymbol{\tau}_t \mid \boldsymbol{\tau}_{<t})$. When an external token has low probability under the policy model, i.e., $\pi_\theta(\boldsymbol{\tau}_t \mid \boldsymbol{\tau}_{<t}) \to 0$, its gradient contribution also approaches zero. This issue is particularly relevant to external tokens that initially have low probability under the policy, including potentially useful reasoning patterns that are uncommon under its current distribution.

Clipping Strategy for External Tokens. To alleviate the vanishing gradient problem, we introduce a lower-bound clipping strategy for external tokens:

$$\text{clip}\left(\pi_\theta, \frac{\omega}{\text{sg}\left(\pi_\theta\right)} \cdot \pi_\theta, \infty\right), \tag{6}$$

where ω is a clipping hyperparameter and $\text{sg}(\cdot)$ denotes the stop-gradient operation. This formulation sets a lower bound on the coefficient for external tokens, while the term $\pi_\theta/\text{sg}(\pi_\theta)$ preserves

numerical equivalence in the forward pass. The resulting gradients are:

$$\begin{cases} \pi_\theta \cdot \tilde{A}_t \cdot \nabla_\theta \log \pi_\theta, & \text{if } \pi_\theta \geq \omega, \\ \omega \cdot \tilde{A}_t \cdot \nabla_\theta \log \pi_\theta, & \text{if } 0 \leq \pi_\theta < \omega. \end{cases} \tag{7}$$

This strategy ensures that low-probability external tokens with positive learning signals still receive non-vanishing gradients, allowing the policy to learn from external feedback even when such tokens initially have low probability.

Final Objective. By integrating the corrected importance sampling coefficient and the proposed clipping strategy, we obtain the final objective of AIPO:

$$\begin{aligned} \widehat{\mathcal{J}}(\theta) = \mathbb{E}_{\pi_{\theta_{\text{old}}}} \left[\frac{1}{|\boldsymbol{\tau}_\iota|} \sum_{\boldsymbol{\tau}_t \in \boldsymbol{\tau}_\iota} \min \left\{ \frac{\pi_\theta^t}{\pi_{\theta_{\text{old}}}^t} \tilde{A}_t, \text{clip} \left(\frac{\pi_\theta^t}{\pi_{\theta_{\text{old}}}^t}, 1-\epsilon, 1+\epsilon \right) \tilde{A}_t \right\} \right] \\ + \mathbb{E}_{\pi_{\theta_{\text{old}}}} \left[\frac{1}{|\boldsymbol{\tau}_\epsilon|} \sum_{\boldsymbol{\tau}_t \in \boldsymbol{\tau}_\epsilon} \text{clip} \left(\pi_\theta^t, \frac{\omega}{\text{sg}\left(\pi_\theta^t\right)} \cdot \pi_\theta^t, \infty \right) \tilde{A}_t \right], \end{aligned} \tag{8}$$

where π_θ^t denotes $\pi_\theta(\boldsymbol{\tau}_t \mid \boldsymbol{\tau}_{<t})$ and $\pi_{\theta_{\text{old}}}^t$ denotes $\pi_{\theta_{\text{old}}}(\boldsymbol{\tau}_t \mid \boldsymbol{\tau}_{<t})$. The first term optimizes internal on-policy tokens using the standard clipped surrogate objective, while the second term enables learning from external off-policy tokens.

4 Experiments

4.1 Setup

Baselines. We compare AIPO against several representative LLM post-training methods: ❶ Supervised Fine-Tuning (SFT): This method fine-tunes the model using synthetic data generated via rejection sampling. ❷ On-Policy Distillation: These methods distill knowledge from a stronger teacher model while preserving on-policy trajectory generation. Specifically, trajectories are sampled from the policy model, whereas the supervision signal is derived from the logits of the teacher model. We adopt OPSD (Zhao et al., 2026) as a representative work. ❸ Reinforcement Learning (RL): These methods train the model using algorithms such as GRPO (Shao et al., 2024), PRIME (Cui et al., 2025), and Dr.GRPO (Liu et al., 2025c). ❹ Off-Policy Enhanced Reinforcement Learning: Represented by LUFFY (Liu et al., 2025b), these methods enhance RL exploration by utilizing trajectories generated by strong models as guidance.

Evaluation Benchmarks. We evaluate all models across four domain-specific benchmarks: ❶ *Mathematical Reasoning*: Includes AIME24, AIME25, MATH500 (Hendrycks et al., 2021), and LiveMathBench (Liu et al., 2024); ❷ *Scientific Reasoning*: Represented by GPQA-Diamond (Rein et al., 2023); ❸ *Code Reasoning*: Comprises MBPP (Austin et al., 2021) and LiveCodeBench (Jain et al., 2025); ❹ *Puzzle Reasoning*: Includes puzzles from Reasoning-Gym (Stojanovski et al., 2025).

Implementation Details. We use Qwen2.5-7B-Instruct (Yang et al., 2024) and Llama3.2-3B-Instruct (Dubey et al., 2024) as policy models, and instantiate collaborators from Qwen2.5-7B-Instruct, Llama3.2-3B-Instruct, or the substantially stronger Qwen3-30B-A3B-Instruct-2507 (Yang et al., 2025). Training uses 34,000 reasoning-intensive examples drawn from DAPO (Yu et al., 2025) and OpenScienceReasoning-2[1] and runs for 200 optimization steps, with at most three collaborator interactions allowed per generation. For all RL-based methods, we evaluate a held-out validation set every 10 training steps and select the checkpoint with the best validation performance. All methods are evaluated under the same prompt templates and sampling configuration. Full training hyperparameters, baseline-specific configurations, evaluation protocol, checkpoint selection, and efficiency statistics are provided in §§ B.1, B.6 to B.8 and C.6.

4.2 Main Results and Analysis

Table 1 illustrates the performance of AIPO and baselines on different benchmarks, with different collaborator LLMs. We summarize the main findings as follows.

AIPO Outperforms Salient Baselines. As shown in Table 1, AIPO achieves superior performance over strong RLVR baselines, demonstrating the effectiveness of our approach in enhancing reasoning

[1] https://huggingface.co/datasets/nvidia/OpenScienceReasoning-2

Table 1: Experimental results of AIPO and baselines with Qwen and Llama, where LUFFY and AIPO are based on GRPO. We report Avg@16 for AIME24 and AIME25 and Avg@4 for the remaining benchmarks, where Avg@k denotes the mean correctness over k independently sampled responses per evaluation problem. The subscripts denote the standard deviation over independent evaluations. We abbreviate LMB as LiveMathBench v202505, LCB as LiveCodeBench v6, and RG as Reasoning Gym. ♠ denotes in-domain evaluation and ♣ denotes out-of-domain evaluation. The performance of Llama is provided in § C.2.

Methods	Math ♠				Science ♠	Code ♣		Puzzle ♣
	AIME24 Avg@16	AIME25 Avg@16	MATH500 Avg@4	LMB Avg@4	GPQA-D Avg@4	MBPP Avg@4	LCB Avg@4	RG Avg@4
Qwen2.5-7B-Instruct								
Original	$9.8_{\pm 0.1}$	$7.5_{\pm 0.3}$	$73.0_{\pm 0.2}$	$10.8_{\pm 0.4}$	$33.3_{\pm 0.1}$	$58.7_{\pm 0.3}$	$15.7_{\pm 0.1}$	$9.6_{\pm 0.2}$
GRPO	$23.3_{\pm 0.4}$	$18.9_{\pm 0.3}$	$78.4_{\pm 0.1}$	$13.9_{\pm 0.2}$	$38.4_{\pm 0.5}$	$61.3_{\pm 0.1}$	$18.1_{\pm 0.4}$	$14.5_{\pm 0.3}$
PRIME	$22.2_{\pm 0.6}$	$18.3_{\pm 0.4}$	$76.5_{\pm 0.3}$	$11.5_{\pm 0.5}$	$35.8_{\pm 0.7}$	$57.7_{\pm 0.4}$	$16.3_{\pm 0.6}$	$12.2_{\pm 0.3}$
Dr.GRPO	$23.7_{\pm 0.3}$	$19.2_{\pm 0.2}$	$78.8_{\pm 0.1}$	$14.0_{\pm 0.1}$	$38.6_{\pm 0.4}$	$61.5_{\pm 0.1}$	$18.4_{\pm 0.3}$	$15.0_{\pm 0.1}$
Qwen2.5-7B-Instruct ↔ *Qwen2.5-7B-Instruct*								
SFT	$20.0_{\pm 0.5}$	$17.1_{\pm 0.4}$	$73.4_{\pm 0.3}$	$11.1_{\pm 0.5}$	$33.6_{\pm 0.7}$	$59.0_{\pm 0.4}$	$15.9_{\pm 0.6}$	$9.8_{\pm 0.3}$
OPSD	$22.8_{\pm 0.4}$	$18.7_{\pm 0.3}$	$77.0_{\pm 0.1}$	$12.8_{\pm 0.3}$	$38.5_{\pm 0.5}$	$60.1_{\pm 0.2}$	$17.0_{\pm 0.4}$	$12.4_{\pm 0.3}$
LUFFY	$23.8_{\pm 0.6}$	$18.4_{\pm 0.5}$	$76.8_{\pm 0.3}$	$13.2_{\pm 0.4}$	$39.1_{\pm 0.6}$	$60.7_{\pm 0.3}$	$17.3_{\pm 0.5}$	$13.9_{\pm 0.4}$
AIPO	$\mathbf{26.5}_{\pm 0.2}$	$\mathbf{21.3}_{\pm 0.1}$	$\mathbf{80.5}_{\pm 0.1}$	$\mathbf{14.9}_{\pm 0.1}$	$\mathbf{41.7}_{\pm 0.3}$	$\mathbf{62.7}_{\pm 0.1}$	$\mathbf{19.2}_{\pm 0.2}$	$\mathbf{16.0}_{\pm 0.1}$
Qwen2.5-7B-Instruct ↔ *Qwen3-30B-A3B-Instruct-2507*								
SFT	$22.4_{\pm 0.5}$	$18.8_{\pm 0.4}$	$76.0_{\pm 0.3}$	$12.5_{\pm 0.5}$	$34.3_{\pm 0.7}$	$59.5_{\pm 0.4}$	$16.4_{\pm 0.6}$	$10.5_{\pm 0.3}$
OPSD	$25.8_{\pm 0.4}$	$20.5_{\pm 0.3}$	$79.2_{\pm 0.1}$	$14.2_{\pm 0.3}$	$40.4_{\pm 0.5}$	$61.3_{\pm 0.2}$	$18.1_{\pm 0.4}$	$14.7_{\pm 0.3}$
LUFFY	$26.7_{\pm 0.6}$	$21.2_{\pm 0.4}$	$80.9_{\pm 0.2}$	$15.1_{\pm 0.4}$	$41.8_{\pm 0.4}$	$62.6_{\pm 0.3}$	$19.2_{\pm 0.5}$	$15.7_{\pm 0.4}$
AIPO	$\mathbf{28.7}_{\pm 0.6}$	$\mathbf{22.4}_{\pm 0.4}$	$\mathbf{82.3}_{\pm 0.2}$	$\mathbf{17.5}_{\pm 0.4}$	$\mathbf{42.9}_{\pm 0.4}$	$\mathbf{63.9}_{\pm 0.3}$	$\mathbf{21.1}_{\pm 0.5}$	$\mathbf{17.8}_{\pm 0.4}$
Llama3.2-3B-Instruct ↔ *Llama3.2-3B-Instruct*								
SFT	$10.6_{\pm 0.5}$	$8.6_{\pm 0.4}$	$61.1_{\pm 0.5}$	$5.1_{\pm 0.5}$	$34.0_{\pm 0.6}$	$38.6_{\pm 0.4}$	$6.2_{\pm 0.5}$	$4.1_{\pm 0.4}$
OPSD	$14.5_{\pm 0.5}$	$9.8_{\pm 0.5}$	$63.2_{\pm 0.4}$	$7.0_{\pm 0.4}$	$35.4_{\pm 0.4}$	$39.8_{\pm 0.5}$	$7.5_{\pm 0.4}$	$4.6_{\pm 0.5}$
LUFFY	$13.6_{\pm 0.6}$	$8.9_{\pm 0.4}$	$62.6_{\pm 0.5}$	$4.8_{\pm 0.6}$	$34.7_{\pm 0.5}$	$39.1_{\pm 0.4}$	$8.7_{\pm 0.4}$	$4.9_{\pm 0.4}$
AIPO	$\mathbf{17.9}_{\pm 0.5}$	$\mathbf{11.9}_{\pm 0.4}$	$\mathbf{67.7}_{\pm 0.5}$	$\mathbf{10.8}_{\pm 0.5}$	$\mathbf{36.4}_{\pm 0.4}$	$\mathbf{43.0}_{\pm 0.4}$	$\mathbf{10.1}_{\pm 0.4}$	$\mathbf{11.0}_{\pm 0.5}$
Llama3.2-3B-Instruct ↔ *Qwen3-30B-A3B-Instruct-2507*								
SFT	$12.4_{\pm 0.4}$	$10.5_{\pm 0.5}$	$62.9_{\pm 0.6}$	$7.0_{\pm 0.4}$	$35.8_{\pm 0.5}$	$40.4_{\pm 0.5}$	$8.0_{\pm 0.4}$	$6.0_{\pm 0.5}$
OPSD	$17.2_{\pm 0.5}$	$11.8_{\pm 0.4}$	$67.0_{\pm 0.5}$	$9.5_{\pm 0.5}$	$35.7_{\pm 0.6}$	$43.5_{\pm 0.4}$	$9.6_{\pm 0.5}$	$10.4_{\pm 0.4}$
LUFFY	$18.7_{\pm 0.4}$	$12.9_{\pm 0.5}$	$67.5_{\pm 0.5}$	$10.8_{\pm 0.6}$	$36.7_{\pm 0.5}$	$44.2_{\pm 0.4}$	$10.8_{\pm 0.5}$	$11.9_{\pm 0.5}$
AIPO	$\mathbf{20.1}_{\pm 0.5}$	$\mathbf{14.4}_{\pm 0.4}$	$\mathbf{69.9}_{\pm 0.5}$	$\mathbf{13.3}_{\pm 0.4}$	$\mathbf{38.9}_{\pm 0.5}$	$\mathbf{45.4}_{\pm 0.3}$	$\mathbf{12.4}_{\pm 0.4}$	$\mathbf{13.3}_{\pm 0.4}$

capabilities. Compared with standard SFT, OPSD, and LUFFY, AIPO obtains consistent improvements across all benchmarks. These results support our claim that active interactions with external collaborators can substantially improve policy exploration and thereby lead to better reasoning performance. Additional efficiency analyses in § C.6 show that these gains persist under a matched computational budget, while quantifying the additional training-time cost introduced by collaborator interaction.

AIPO Generalizes to Different Policy Models. The benefits of AIPO are not tied to a specific model architecture, but generalize across different foundation models. When applied to both Qwen2.5-7B-Instruct and Llama3.2-3B-Instruct, AIPO yields consistent improvements over the LUFFY baseline, while additional RLVR results on Llama3.2-3B-Instruct are reported in § C.2. We further evaluate AIPO on the larger Qwen2.5-32B-Instruct model and the long-CoT DeepSeek-R1-Distill-Qwen-1.5B model in §§ C.3 and C.5, respectively, demonstrating consistent effectiveness across model scales and reasoning behaviors. Notably, even when the policy model and collaborators share the same backbone, AIPO still outperforms standard RL algorithms. This indicates that the gains of AIPO do not simply come from using a stronger external model, but are primarily driven by enhanced exploration through active interaction.

AIPO Generalizes to Different RLVR Algorithms. To evaluate the generalization capability of AIPO across different RLVR algorithms, we further implement AIPO with both DAPO (Yu et al., 2025) and GSPO (Zheng et al., 2025). The results, presented in § C.4, show that AIPO delivers consistent performance improvements regardless of the underlying RLVR algorithm, thereby demonstrating its strong generalizability.

Table 2: Ablation study of AIPO's components w.r.t. the corrected importance sampling coefficient (abbreviated as IS, Equation (5)) and clipping strategy (abbreviated as CS, Equation (6)). We also include the results where we masked the external tokens generated by collaborators.

Methods	Math ♠				Science ♠	Code ♣		Puzzle ♣
	AIME24 Avg@16	AIME25 Avg@16	MATH500 Avg@4	LMB Avg@4	GPQA-D Avg@4	MBPP Avg@4	LCB Avg@4	RG Avg@4
AIPO	**26.5**$_{\pm 0.2}$	**21.3**$_{\pm 0.1}$	**80.5**$_{\pm 0.1}$	**14.9**$_{\pm 0.1}$	**41.7**$_{\pm 0.3}$	**62.7**$_{\pm 0.1}$	**19.2**$_{\pm 0.2}$	**16.0**$_{\pm 0.1}$
w/o IS	25.6$_{\pm 0.4}\downarrow 0.9$	20.5$_{\pm 0.3}\downarrow 0.8$	77.5$_{\pm 0.5}\downarrow 3.0$	13.7$_{\pm 0.5}\downarrow 1.2$	40.0$_{\pm 0.5}\downarrow 1.7$	61.6$_{\pm 0.4}\downarrow 1.1$	16.5$_{\pm 0.6}\downarrow 2.7$	15.1$_{\pm 0.4}\downarrow 0.9$
w/o CS	25.5$_{\pm 0.4}\downarrow 1.0$	19.8$_{\pm 0.5}\downarrow 1.5$	76.9$_{\pm 0.6}\downarrow 3.6$	10.6$_{\pm 0.4}\downarrow 4.3$	38.6$_{\pm 0.5}\downarrow 3.1$	60.3$_{\pm 0.5}\downarrow 2.4$	15.4$_{\pm 0.5}\downarrow 3.8$	14.2$_{\pm 0.5}\downarrow 1.8$
Mask	25.0$_{\pm 0.5}\downarrow 1.5$	19.4$_{\pm 0.4}\downarrow 1.9$	79.7$_{\pm 0.3}\downarrow 0.8$	11.9$_{\pm 0.5}\downarrow 3.0$	37.8$_{\pm 0.6}\downarrow 3.9$	59.7$_{\pm 0.4}\downarrow 3.0$	15.1$_{\pm 0.5}\downarrow 4.1$	13.4$_{\pm 0.5}\downarrow 2.6$

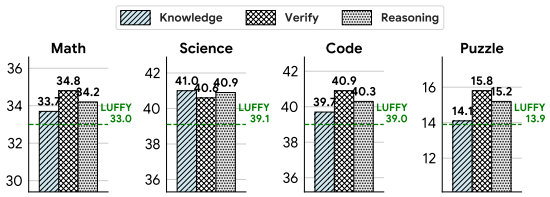

Figure 3: Ablation Study of the collaborators in AIPO. Each bar indicates the average performance of all benchmarks in this domain.

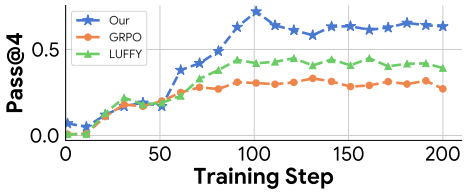

Figure 4: Expansion of the effective exploration boundary on DeepMath, measured by Pass@4.

AIPO Generalizes to Out-of-Domain Benchmarks. Beyond standard mathematical and scientific reasoning tasks, AIPO exhibits strong generalization across distinct domains. The results show consistent improvements over baselines on code generation benchmarks, including MBPP and LCB, as well as complex puzzle tasks in RG. This indicates that the enhanced exploration enabled by AIPO transfers beyond the training distribution and benefits broader reasoning scenarios.

AIPO Benefits from Stronger Collaborators. AIPO can effectively exploit additional information provided by a stronger collaborator. Replacing same-backbone collaborators with Qwen3-30B-A3B-Instruct-2507, whose standalone performance is reported in Table 4, improves the overall performance of both policy-model families. These results indicate that collaborator capability is one factor affecting the usefulness of interactive exploration. However, the benefit is not necessarily monotonic: once collaborator feedback exceeds what the policy can effectively internalize, further increases in collaborator capability need not translate proportionally into policy improvement. We analyze this effect further in § C.9.

4.3 Ablation Study

We ablate the three components of AIPO in Table 2 and Figure 3. Replacing the corrected importance sampling coefficient (Equation (5)) with the standard ratio computed directly under the policy model consistently degrades performance, indicating that it is an effective approximation of the policy–collaborator discrepancy and mitigates off-policy bias. Removing the clipping strategy (Equation (6)) causes a substantial drop across all benchmarks. Complementary diagnostics in § C.8 show that a substantial fraction of external tokens initially fall into the near-vanishing regime, directly motivating the proposed lower-bound clipping. Restricting the policy to a single collaborator reveals a clear division of labour: the Knowledge Agent contributes most on knowledge-intensive benchmarks such as science, while the Verify and Reasoning Agents are more effective elsewhere. Nevertheless, any single collaborator still outperforms the baseline. Complementary usage statistics in § C.7 show that the learned policy does not collapse onto a single collaborator, but continues to make substantial use of all three agent types during training.

4.4 Does AIPO Expand the Effective Exploration Boundary?

We examine what AIPO actually learns through its Pass@4 behaviour on DeepMath (He et al., 2025b), which is disjoint from our training corpus, and through its training dynamics. As shown in Figure 4, all three methods are indistinguishable for the first $\sim$50 steps, after which AIPO separates sharply, peaks around step 100, and then stabilises at roughly 0.63, against 0.42 for LUFFY and 0.30 for GRPO. Figure 5 traces the underlying mechanism. AIPO drives the number of tasks failed across *all* rollouts in a batch (*Batch Failed Tasks*) from $\sim$150 down to $\sim$45, whereas GRPO stalls near 130, and batch accuracy rises to $\sim$0.48 against $\sim$0.26 for GRPO. Interaction frequency (*Batch*

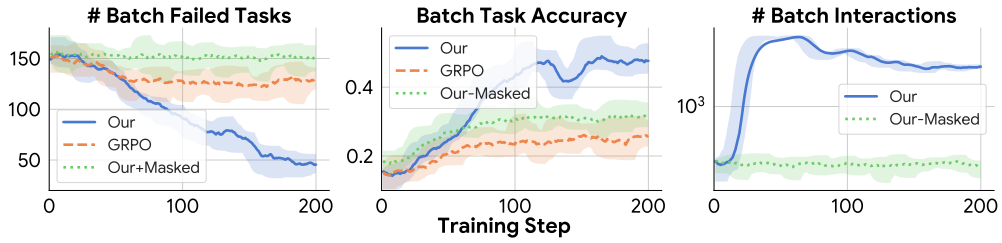

Figure 5: Training Dynamics of AIPO and baselines on Qwen2.5-7B-Instruct with the same model as collaborators.

Interactions) climbs steeply over the first $\sim$60 steps, then declines and stabilises, consistent with a policy that relies more heavily on collaborators early in training and increasingly solves problems independently as it internalises their guidance. Detailed invocation statistics in § C.7 further show that approximately 10% of training samples require no interaction and that the policy continues to use all three collaborator types rather than collapsing onto a single agent. Masking external tokens severs exactly this internalisation: interaction frequency never leaves its initial level, and although batch accuracy still improves to ~ 0.31, *Batch Failed Tasks* remains flat at ~ 150 throughout training. The masked variant therefore gets better at problems it could already solve while never cracking the ones it could not, which is precisely the difference between consuming collaborator information and integrating it into the policy's own parameters (Table 2).

5 Related Work

Reinforcement Learning for LLM Reasoning. Reinforcement Learning with Verifiable Rewards (RLVR) has become a widely used paradigm for improving LLM reasoning (OpenAI, 2024b; 2025; DeepSeek-AI et al., 2025; Kimi-Team et al., 2025; Yang et al., 2025; Comanici et al., 2025). Recent methods further improve RLVR through advances in sampling, reward design, and policy optimization, including Dr.GRPO (Liu et al., 2025c), VAPO (Yue et al., 2025b), and DAPO (Yu et al., 2025). Despite these advances, purely on-policy RLVR remains limited by the reasoning trajectories that the current policy can discover during rollout (Zhao et al., 2025; Yue et al., 2025a).

External-Guidance-Enhanced Learning. A growing line of work introduces external supervision through expert trajectories (Zhang et al., 2025c; Lv et al., 2025; Zhang et al., 2025b; Chen et al., 2025b; Sun et al., 2025b), teacher distributions (Xu et al., 2025; Zhao et al., 2026), critiques (Shinn et al., 2023; Lin et al., 2025), or supervised expert data (Ma et al., 2025; Fu et al., 2025). In contrast to methods that provide supervision over complete trajectories or policy-generated rollouts, AIPO introduces targeted collaborator feedback during rollout, producing mixed-policy trajectories conditioned on the policy's current reasoning bottlenecks. The collaborators are used only during training, while the resulting policy performs inference independently. Detailed comparisons with on-policy distillation, off-policy expert learning, multi-agent guidance, and self-reflection methods are provided in § D.1.

6 Conclusion

In this paper, we introduce **AIPO**, an enhanced RLVR framework that broadens policy exploration through active interaction with external collaborators during training. Rather than relying on complete expert trajectories, AIPO allows the policy to selectively request verification, knowledge, or sub-problem reasoning when it encounters intermediate bottlenecks, producing mixed-policy trajectories with fine-grained and state-dependent guidance. To learn effectively from the externally generated tokens in these trajectories, we introduce an corrected importance sampling coefficient and a lower-bound clipping strategy that mitigate off-policy discrepancy and vanishing gradients. Across diverse mathematical, scientific, coding, and puzzle benchmarks, AIPO consistently improves over representative RLVR, distillation, and expert-guided baselines and generalizes across policy models, collaborator backbones, and RLVR algorithms. Further analyses indicate that training-time interaction broadens the policy's effective exploration boundary, while the trained policy operates independently without collaborators or additional inference-time computation.

AI USE STATEMENT

We use generative AI tools to refine the writing of this paper for readability and use AI coding assistance for vibing code development. No other disclosure categories apply. We have reviewed all AI-assisted text and take full responsibility for the final content of this work, including all claims and artifacts.

ETHICS STATEMENT

We focus on general scientific research questions and pose no risks to safety, personal security, or privacy. No new datasets are released as part of this study. Additionally, the research does not encompass potentially harmful insights, methods, or applications, nor does it raise issues related to privacy, security, legal compliance, or research integrity. We foresee no ethical risks or conflicts of interest. We are committed to adhering to ethical guidelines throughout the research process.

REPRODUCIBILITY STATEMENT

We provide a comprehensive description of the proposed AIPO in § 3, with detailed implementation specifics provided in § B and § 4.1. All datasets utilized in this research are publicly available. Key code implementations are included in the supplementary materials for reference, and the complete code will be made publicly available upon acceptance of the paper.

REFERENCES

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms. In *ACL (1)*, pp. 12248–12267. Association for Computational Linguistics, 2024. 2

Jacob Austin, Augustus Odena, Maxwell I. Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie J. Cai, Michael Terry, Quoc V. Le, and Charles Sutton. Program synthesis with large language models. *CoRR*, abs/2108.07732, 2021. 1, 4.1, B.10

Shelly Bensal, Umar Jamil, Christopher Bryant, Melisa Russak, Kiran Kamble, Dmytro Mozolevskyi, Muayad Ali, and Waseem AlShikh. Reflect, retry, reward: Self-improving llms via reinforcement learning. *CoRR*, abs/2505.24726, 2025. D.2

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. *CoRR*, abs/2509.25666, 2025a. 1

Liang Chen, Xueting Han, Li Shen, Jing Bai, and Kam-Fai Wong. Beyond two-stage training: Cooperative SFT and RL for LLM reasoning. *CoRR*, abs/2509.06948, 2025b. 1, 2, 5, D.1

Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *CoRR*, abs/2503.09567, 2025c. 1

Yixing Chen, Yiding Wang, Siqi Zhu, Haofei Yu, Tao Feng, Muhan Zhang, Mostofa Patwary, and Jiaxuan You. Multi-agent evolve: LLM self-improve through co-evolution. *CoRR*, abs/2510.23595, 2025d. 1

Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Wayne Xin Zhao, Zhenliang Zhang, and Furu Wei. Reasoning with exploration: An entropy perspective. *CoRR*, abs/2506.14758, 2025. A

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit S. Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, Krishna Haridasan, Ahmed Omran, Nikunj Saunshi, Dara Bahri, Gaurav Mishra, Eric

Chu, Toby Boyd, Brad Hekman, Aaron Parisi, Chaoyi Zhang, Kornraphop Kawintiranon, Tania Bedrax-Weiss, Oliver Wang, Ya Xu, Ollie Purkiss, Uri Mendlovic, Ilaï Deutel, Nam Nguyen, Adam Langley, Flip Korn, Lucia Rossazza, Alexandre Ramé, Sagar Waghmare, Helen Miller, Nathan Byrd, Ashrith Sheshan, Raia Hadsell Sangnie Bhardwaj, Pawel Janus, Tero Rissa, Dan Horgan, Sharon Silver, Ayzaan Wahid, Sergey Brin, Yves Raimond, Klemen Kloboves, Cindy Wang, Nitesh Bharadwaj Gundavarapu, Ilia Shumailov, Bo Wang, Mantas Pajarskas, Joe Heyward, Martin Nikoltchev, Maciej Kula, Hao Zhou, Zachary Garrett, Sushant Kafle, Sercan Arik, Ankita Goel, Mingyao Yang, Jiho Park, Koji Kojima, Parsa Mahmoudieh, Koray Kavukcuoglu, Grace Chen, Doug Fritz, Anton Bulyenov, Sudeshna Roy, Dimitris Paparas, Hadar Shemtov, Bo-Juen Chen, Robin Strudel, David Reitter, Aurko Roy, Andrey Vlasov, Changwan Ryu, Chas Leichner, Haichuan Yang, Zelda Mariet, Denis Vnukov, Tim Sohn, Amy Stuart, Wei Liang, Minmin Chen, Praynaa Rawlani, Christy Koh, JD Co-Reyes, Guangda Lai, Praseem Banzal, Dimitrios Vytiniotis, Jieru Mei, and Mu Cai. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *CoRR*, abs/2507.06261, 2025. 1, 2, 5

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, Hao Peng, Yu Cheng, Zhiyuan Liu, Maosong Sun, Bowen Zhou, and Ning Ding. Process reinforcement through implicit rewards. *CoRR*, abs/2502.01456, 2025. 4.1

Xingyu Dang, Christina Baek, Kaiyue Wen, Zico Kolter, and Aditi Raghunathan. Weight ensembling improves reasoning in language models. *CoRR*, abs/2504.10478, 2025. 1

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, and S. S. Li. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948, 2025. 1, 2, 5, B.6, C.5, D.2

Yihe Deng, I-Hung Hsu, Jun Yan, Zifeng Wang, Rujun Han, Gufeng Zhang, Yanfei Chen, Wei Wang, Tomas Pfister, and Chen-Yu Lee. Supervised reinforcement learning: From expert trajectories to step-wise reasoning. *CoRR*, abs/2510.25992, 2025. 1

Zi-Yi Dou, Cheng-Fu Yang, Xueqing Wu, Kai-Wei Chang, and Nanyun Peng. Re-rest: Reflection-reinforced self-training for language agents. In *EMNLP*, pp. 15394–15411. Association for Computational Linguistics, 2024. D.1

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi,

Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. The llama 3 herd of models. *CoRR*, abs/2407.21783, 2024. 1, 4.1

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. *CoRR*, abs/2506.19767, 2025. 1, 1, 2, 5, D.1

Hongcheng Gao, Yue Liu, Yufei He, Longxu Dou, Chao Du, Zhijie Deng, Bryan Hooi, Min Lin, and Tianyu Pang. Flowreasoner: Reinforcing query-level meta-agents. *CoRR*, abs/2504.15257, 2025. 3.1

Etash Kumar Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, Ashima Suvarna, Benjamin Feuer, Liangyu Chen, Zaid Khan, Eric Frankel, Sachin Grover, Caroline Choi, Niklas Muennighoff, Shiye Su, Wanjia Zhao, John Yang, Shreyas Pimpalgaonkar, Kartik Sharma, Charlie Cheng-Jie Ji, Yichuan Deng, Sarah M. Pratt, Vivek Ramanujan, Jon Saad-Falcon, Jeffrey Li, Achal Dave, Alon Albalak, Kushal Arora, Blake Wulfe, Chinmay Hegde, Greg Durrett, Sewoong Oh, Mohit Bansal, Saadia Gabriel, Aditya Grover, Kai-Wei Chang, Vaishaal Shankar, Aaron Gokaslan, Mike A. Merrill, Tatsunori Hashimoto, Yejin Choi, Jenia Jitsev, Reinhard Heckel, Maheswaran Sathiamoorthy, Alexandros G. Dimakis, and Ludwig Schmidt. Openthoughts: Data recipes for reasoning models. *CoRR*, abs/2506.04178, 2025. 3.1

Andre Wang He, Daniel Fried, and Sean Welleck. Rewarding the unlikely: Lifting GRPO beyond distribution sharpening. In *EMNLP*, pp. 25548–25560. Association for Computational Linguistics, 2025a. 1

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. *CoRR*, abs/2504.11456, 2025b. 4.4

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. In *NeurIPS Datasets and Benchmarks*, 2021. 1, 4.1, B.10

Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B. Brown, Prafulla Dhariwal, Scott Gray, Chris Hallacy, Benjamin Mann, Alec Radford, Aditya Ramesh, Nick Ryder, Daniel M. Ziegler, John Schulman, Dario Amodei, and Sam McCandlish. Scaling laws for autoregressive generative modeling. *CoRR*, abs/2010.14701, 2020. E

Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory F. Diamos, Heewoo Jun, Hassan Kianinejad, Md. Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically. *CoRR*, abs/1712.00409, 2017. E

Jian Hu. REINFORCE++: A simple and efficient approach for aligning large language models. *CoRR*, abs/2501.03262, 2025. 2

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *CoRR*, abs/2503.24290, 2025. 2

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.*, 43 (2):42:1–42:55, 2025. D.2

Yichen Huang and Lin F. Yang. Gemini 2.5 pro capable of winning gold at IMO 2025. *CoRR*, abs/2507.15855, 2025. 1

Zhen Huang, Haoyang Zou, Xuefeng Li, Yixiu Liu, Yuxiang Zheng, Ethan Chern, Shijie Xia, Yiwei Qin, Weizhe Yuan, and Pengfei Liu. O1 replication journey - part 2: Surpassing o1-preview through simple distillation, big progress or bitter lesson? *CoRR*, abs/2411.16489, 2024. 3.1

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *ICLR*. OpenReview.net, 2025. 1, 4.1, B.10

Bowen Jin, Hansi Zeng, Zhenrui Yue, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *CoRR*, abs/2503.09516, 2025. 3.2

Kimi-Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, Chuning Tang, Congcong Wang, Dehao Zhang, Enming Yuan, Enzhe Lu, Fengxiang Tang, Flood Sung, Guangda Wei, Guokun Lai, Haiqing Guo, Han Zhu, Hao Ding, Hao Hu, Hao Yang, Hao Zhang, Haotian Yao, Haotian Zhao, Haoyu Lu, Haoze Li, Haozhen Yu, Hongcheng Gao, Huabin Zheng, Huan Yuan, Jia Chen, Jianhang Guo, Jianlin Su, Jianzhou Wang, Jie Zhao, Jin Zhang, Jingyuan Liu, Junjie Yan, Junyan Wu, Lidong Shi, Ling Ye, Longhui Yu, Mengnan Dong, Neo Zhang, Ningchen Ma, Qiwei Pan, Qucheng Gong, Shaowei Liu, Shengling Ma, Shupeng Wei, Sihan Cao, Siying Huang, Tao Jiang, Weihao Gao, Weimin Xiong, Weiran He, Weixiao Huang, Wenhao Wu, Wenyang He, Xianghui Wei, Xianqing Jia, Xingzhe Wu, Xinran Xu, Xinxing Zu, Xinyu Zhou, Xuehai Pan, Y. Charles, Yang Li, Yangyang Hu, Yangyang Liu, Yanru Chen, Yejie Wang, Yibo Liu, Yidao Qin, Yifeng Liu, Ying Yang, Yiping Bao, Yulun Du, Yuxin Wu, Yuzhi Wang, Zaida Zhou, Zhaoji Wang, Zhaowei Li, Zhen Zhu, Zheng Zhang, Zhexu Wang, Zhilin Yang, Zhiqi Huang, Zihao Huang, Ziyao Xu, and Zonghan Yang. Kimi k1.5: Scaling reinforcement learning with llms. *CoRR*, abs/2501.12599, 2025. 1, 2, 5

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D. Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, Lei M. Zhang, Kay McKinney, Disha Shrivastava, Cosmin Paduraru, George Tucker, Doina Precup, Feryal M. P. Behbahani, and Aleksandra Faust. Training language models to self-correct via reinforcement learning. In *ICLR*. OpenReview.net, 2025. D.2

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *SOSP*, pp. 611–626. ACM, 2023. B.1

Ziniu Li, Tian Xu, Yushun Zhang, Zhihang Lin, Yang Yu, Ruoyu Sun, and Zhi-Quan Luo. Remax: A simple, effective, and efficient reinforcement learning method for aligning large language models. In *ICML*. OpenReview.net, 2024. 2

Junwei Liao, Muning Wen, Jun Wang, and Weinan Zhang. MARFT: multi-agent reinforcement fine-tuning. *CoRR*, abs/2504.16129, 2025a. 3.1

Mengqi Liao, Xiangyu Xi, Ruinian Chen, Jia Leng, Yangen Hu, Ke Zeng, Shuai Liu, and Huaiyu Wan. Enhancing efficiency and exploration in reinforcement learning for llms. In *EMNLP*, pp. 1451–1463. Association for Computational Linguistics, 2025b. 1

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *ICLR*. OpenReview.net, 2024. B.10

Hehai Lin, Shilei Cao, Sudong Wang, Haotian Wu, Minzhi Li, Linyi Yang, Juepeng Zheng, and Chengwei Qin. Interactive learning for LLM reasoning. *CoRR*, abs/2509.26306, 2025. 1, 5, D.1

Junnan Liu, Hongwei Liu, Linchen Xiao, Ziyi Wang, Kuikun Liu, Songyang Gao, Wenwei Zhang, Songyang Zhang, and Kai Chen. Are your llms capable of stable reasoning? *CoRR*, abs/2412.13147, 2024. 1, 4.1, B.10

Junnan Liu, Linhao Luo, Thuy-Trang Vu, and Gholamreza Haffari. Situatedthinker: Grounding LLM reasoning with real-world through situated thinking. *CoRR*, abs/2505.19300, 2025a. 3.2

Wei Liu, Ruochen Zhou, Yiyun Deng, Yuzhen Huang, Junteng Liu, Yuntian Deng, Yizhe Zhang, and Junxian He. Learn to reason efficiently with adaptive length-based reward shaping. *CoRR*, abs/2505.15612, 2025b. 4.1

Zeyuan Liu, Jeonghye Kim, Xufang Luo, Dongsheng Li, and Yuqing Yang. Exploratory memory-augmented llm agent via hybrid on- and off-policy optimization. *CoRR*, abs/2602.23008, 2026. 1

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. *CoRR*, abs/2503.20783, 2025c. 4.1, 5

Xingtai Lv, Yuxin Zuo, Youbang Sun, Hongyi Liu, Yuntian Wei, Zhekai Chen, Lixuan He, Xuekai Zhu, Kaiyan Zhang, Bingning Wang, Ning Ding, and Bowen Zhou. Towards a unified view of large language model post-training. *CoRR*, abs/2509.04419, 2025. 1, 2, 5, D.1

Lu Ma, Hao Liang, Meiyi Qiang, Lexiang Tang, Xiaochen Ma, Zhen Hao Wong, Junbo Niu, Chengyu Shen, Runming He, Bin Cui, and Wentao Zhang. Learning what reinforcement learning can't: Interleaved online fine-tuning for hardest questions. *CoRR*, abs/2506.07527, 2025. 2, 5, D.1

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel J. Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. *CoRR*, abs/2501.19393, 2025. 3.1

OpenAI. Learning to reason with llms. https://openai.com/index/learning-to-reason-with-llms/, 2024a. Accessed: 2024-09. 1, 2

OpenAI. Introducing openai o3 and o4-mini. https://openai.com/index/introducing-o3-and-o4-mini/, 2024b. Accessed: 2024-12. 1, 2, 5

OpenAI. Gpt-5 and the new era of work. https://openai.com/index/gpt-5-new-era-of-work/, 2025. Accessed: 2025-08. 1, 5

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *NeurIPS*, 2022. 2

Yiwei Qin, Xuefeng Li, Haoyang Zou, Yixiu Liu, Shijie Xia, Zhen Huang, Yixin Ye, Weizhe Yuan, Hector Liu, Yuanzhi Li, and Pengfei Liu. O1 replication journey: A strategic progress report - part 1. *CoRR*, abs/2410.18982, 2024. 3.1

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. *CoRR*, abs/2311.12022, 2023. 1, 4.1, B.10

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. 1, 2, 3.2

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300, 2024. 1, 1, 2, 4.1

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 *EuroSys*, pp. 1279–1297. ACM, 2025. B.1

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In *NeurIPS*, 2023. 5, D.1

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *CoRR*, abs/2408.03314, 2024. 1

Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. R1-searcher: Incentivizing the search capability in llms via reinforcement learning. *CoRR*, abs/2503.05592, 2025. 3.2

Zafir Stojanovski, Oliver Stanley, Joe Sharratt, Richard Jones, Abdulhakeem Adefioye, Jean Kaddour, and Andreas Köpf. REASONING GYM: reasoning environments for reinforcement learning with verifiable rewards. *CoRR*, abs/2505.24760, 2025. 1, 4.1, B.10

Hao Sun, Zile Qiao, Jiayan Guo, Xuanbo Fan, Yingyan Hou, Yong Jiang, Pengjun Xie, Yan Zhang, Fei Huang, and Jingren Zhou. Zerosearch: Incentivize the search capability of llms without searching. *CoRR*, abs/2505.04588, 2025a. 3.1, B.9

Yifan Sun, Jingyan Shen, Yibin Wang, Tianyu Chen, Zhendong Wang, Mingyuan Zhou, and Huan Zhang. Improving data efficiency for LLM reinforcement fine-tuning through difficulty-targeted online data selection and rollout replay. *CoRR*, abs/2506.05316, 2025b. 5, D.1

Ziyu Wan, Yunxiang Li, Yan Song, Hanjing Wang, Linyi Yang, Mark Schmidt, Jun Wang, Weinan Zhang, Shuyue Hu, and Ying Wen. Rema: Learning to meta-think for llms with multi-agent reinforcement learning. *CoRR*, abs/2503.09501, 2025. 3.1

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for LLM reasoning. *CoRR*, abs/2506.01939, 2025. A

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, 2022. 3.1

Zhepei Wei, Xiao Yang, Kai Sun, Jiaqi Wang, Rulin Shao, Sean Chen, Mohammad Kachuee, Teja Gollapudi, Tony Liao, Nicolas Scheffer, Rakesh Wanga, Anuj Kumar, Yu Meng, Wen-tau Yih, and Xin Luna Dong. Truthrl: Incentivizing truthful llms via reinforcement learning. *CoRR*, abs/2509.25760, 2025. D.2

xAI. Grok 4. https://x.ai/news/grok-4/, 2025. Accessed: 2025-07. 1

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. *CoRR*, abs/2506.02208, 2025. 5

Xiangyuan Xue, Yifan Zhou, Guibin Zhang, Zaibin Zhang, Yijiang Li, Chen Zhang, Zhenfei Yin, Philip Torr, Wanli Ouyang, and Lei Bai. Comas: Co-evolving multi-agent systems via interaction rewards. *CoRR*, abs/2510.08529, 2025. 1

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *CoRR*, abs/2412.15115, 2024. 1, 4.1, C.3

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jian Yang, Jiaxi Yang, Jingren Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *CoRR*, abs/2505.09388, 2025. 1, 2, 4.1, 5, B.6

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Weinan Dai, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. DAPO: an open-source LLM reinforcement learning system at scale. *CoRR*, abs/2503.14476, 2025. 1, 4.1, 4.2, 5, B.5, C.4

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? *CoRR*, abs/2504.13837, 2025a. 1, 5, E

Yu Yue, Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Cheng-Xiang Wang, Tiantian Fan, Zhengyin Du, Xiangpeng Wei, Xiangyu Yu, Gaohong Liu, Juncai Liu, Lingjun Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Ru Zhang, Xin Liu, Mingxuan Wang, Yonghui Wu, and Lin Yan. VAPO: efficient and reliable reinforcement learning for advanced reasoning tasks. *CoRR*, abs/2504.05118, 2025b. 5

Fuxiang Zhang, Jiacheng Xu, Chaojie Wang, Ce Cui, Yang Liu, and Bo An. Incentivizing llms to self-verify their answers. *CoRR*, abs/2506.01369, 2025a. D.2

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. *CoRR*, abs/2508.11408, 2025b. 1, 2, 2, 5, D.1

Yue Zhang, Yafu Li, Ganqu Cui, Yu Cheng, Zhi Wang, Xiaoye Qu, Jianhao Yan, and Zican Hu. Learning to reason under off-policy guidance. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025c. 1, 2, 2, 3.2, 5, D.1

Rosie Zhao, Alexandru Meterez, Sham M. Kakade, Cengiz Pehlevan, Samy Jelassi, and Eran Malach. Echo chamber: RL post-training amplifies behaviors learned in pretraining. *CoRR*, abs/2504.07912, 2025. 1, 5

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *CoRR*, abs/2601.18734, 2026. 1, 4.1, 5, D.1

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization. *CoRR*, abs/2507.18071, 2025. 1, 4.2, C.4

Appendix

Table of Contents

A Analysis of Approximation Error of Corrected Importance Sampling Coefficient **18**

B More Implementation Details **19**
- B.1 Training Details . . . 19
- B.2 Full Training Prompt . . . 19
- B.3 Collaborator Prompt . . . 20
- B.4 Inference Prompt . . . 23
- B.5 Training Data . . . 23
- B.6 Evaluation Protocol . . . 23
- B.7 Checkpoint Selection . . . 24
- B.8 Baseline Implementation Details . . . 24
- B.9 Design of the Knowledge Agent Prompt . . . 24
- B.10 Evaluation Benchmarks . . . 25

C Additional Experimental Results and Analysis **28**
- C.1 Performance of External LLMs . . . 28
- C.2 Performance of Reinforcement Learning on Llama3.2-3B-Instruct . . . 28
- C.3 AIPO on Larger Policy Models . . . 29
- C.4 AIPO on Different RLVR Algorithms . . . 29
- C.5 AIPO on Long-CoT Policy Models . . . 29
- C.6 Efficiency Analysis . . . 30
- C.7 Statistics of Collaborator Invocation . . . 31
- C.8 Proportion of Near-Vanishing Gradients . . . 32
- C.9 Effect of Collaborator Capability . . . 32
- C.10 Case Study . . . 32

D Discussions **36**
- D.1 More Discussions on Related Work . . . 36
- D.2 Mitigating Errors in Active Interactions . . . 37
- D.3 Why AIPO Does Not Require Distributionally Matched Collaborators . . . 37
- D.4 Comparison with On-Policy Distillation . . . 38
- D.5 Can the Policy Model Surpass its Collaborators? . . . 38
- D.6 Comparison with SFT Initialization followed by RL . . . 38

E Limitations and Future Work **39**

A Analysis of Approximation Error of Corrected Importance Sampling Coefficient

In this section, we analyze the approximation error introduced by the corrected importance sampling coefficient defined in Equation (5), compared with the original importance sampling coefficient. Our key idea is to decompose the approximation error into two terms: the concentration of the current policy distribution and the total variation distance between the current and old policies.

Proposition A.1. *For any policies π_θ, $\pi_{\theta_{old}}$, and π_ϵ, suppose that the advantage is bounded as $|\tilde{A}_t| \leq A_{\max}$. Define the concentration term of the current policy as*

$$\eta(\pi_\theta) = 1 - \sum_{\tau_t} \pi_\theta(\tau_t \mid \tau_{<t})^2. \tag{9}$$

Then, the approximation error satisfies

$$\begin{aligned} &\left| \mathbb{E}_{\theta_{old}} \left[\sum_{\tau_t \in \tau_\epsilon} \pi_\theta(\tau_t \mid \tau_{<t}) \tilde{A}_t \right] - \mathbb{E}_{\pi_\epsilon} \left[\sum_{\tau_t \in \tau_\epsilon} \frac{\pi_\theta(\tau_t \mid \tau_{<t})}{\pi_\epsilon(\tau_t \mid I_t)} \tilde{A}_t \right] \right| \\ &\quad \leq A_{\max} \left(\eta(\pi_\theta) + 2 d_{\mathrm{TV}} \left(\pi_\theta, \pi_{\theta_{old}} \right) \right), \end{aligned} \tag{10}$$

where d_{TV} denotes the total variation distance:

$$d_{\mathrm{TV}} \left(\pi_\theta, \pi_{\theta_{old}} \right) = \frac{1}{2} \sum_{\tau_t} \left| \pi_\theta(\tau_t \mid \tau_{<t}) - \pi_{\theta_{old}}(\tau_t \mid \tau_{<t}) \right|. \tag{11}$$

Proof. Expanding the true importance-weighted term gives

$$\begin{aligned} \mathbb{E}_{\pi_\epsilon} \left[\frac{\pi_\theta(\tau_t \mid \tau_{<t})}{\pi_\epsilon(\tau_t \mid I_t)} \tilde{A}_t \right] &= \sum_{\tau_t} \pi_\epsilon(\tau_t \mid I_t) \frac{\pi_\theta(\tau_t \mid \tau_{<t})}{\pi_\epsilon(\tau_t \mid I_t)} \tilde{A}_t \\ &= \sum_{\tau_t} \pi_\theta(\tau_t \mid \tau_{<t}) \tilde{A}_t. \end{aligned} \tag{12}$$

Meanwhile, the corrected term under $\pi_{\theta_{\text{old}}}$ can be written as

$$\mathbb{E}_{\theta_{\text{old}}} \left[\pi_\theta(\tau_t \mid \tau_{<t}) \tilde{A}_t \right] = \sum_{\tau_t} \pi_{\theta_{\text{old}}}(\tau_t \mid \tau_{<t}) \pi_\theta(\tau_t \mid \tau_{<t}) \tilde{A}_t. \tag{13}$$

Therefore, the approximation error is

$$\begin{aligned} \Delta &= \left| \sum_{\tau_t} \pi_\theta(\tau_t \mid \tau_{<t}) \tilde{A}_t - \sum_{\tau_t} \pi_{\theta_{\text{old}}}(\tau_t \mid \tau_{<t}) \pi_\theta(\tau_t \mid \tau_{<t}) \tilde{A}_t \right| \\ &= \left| \sum_{\tau_t} \pi_\theta(\tau_t \mid \tau_{<t}) \left(1 - \pi_{\theta_{\text{old}}}(\tau_t \mid \tau_{<t}) \right) \tilde{A}_t \right|. \end{aligned} \tag{14}$$

Using $|\tilde{A}_t| \leq A_{\max}$, we obtain

$$\Delta \leq A_{\max} \sum_{\tau_t} \pi_\theta(\tau_t \mid \tau_{<t}) \left(1 - \pi_{\theta_{\text{old}}}(\tau_t \mid \tau_{<t}) \right). \tag{15}$$

The summation can be rewritten as

$$\begin{aligned} \sum_{\tau_t} \pi_\theta \left(1 - \pi_{\theta_{\text{old}}} \right) &= 1 - \langle \pi_\theta, \pi_{\theta_{\text{old}}} \rangle \\ &= 1 - \|\pi_\theta\|_2^2 + \langle \pi_\theta, \pi_\theta - \pi_{\theta_{\text{old}}} \rangle \\ &\leq \eta(\pi_\theta) + \|\pi_\theta\|_\infty \|\pi_\theta - \pi_{\theta_{\text{old}}}\|_1 \\ &\leq \eta(\pi_\theta) + 2 d_{\mathrm{TV}} \left(\pi_\theta, \pi_{\theta_{\text{old}}} \right), \end{aligned} \tag{16}$$

where $\eta(\pi_\theta) = 1 - \|\pi_\theta\|_2^2$ and $\|\pi_\theta\|_\infty \leq 1$. Combining the above inequalities yields

$$\Delta \leq A_{\max} \left(\eta(\pi_\theta) + 2 d_{\mathrm{TV}} \left(\pi_\theta, \pi_{\theta_{\text{old}}} \right) \right). \tag{17}$$

This completes the proof. □

Theorem A.1 shows that the approximation error is controlled by two factors: the concentration of the current policy distribution and the total variation distance between the current and old policies. The first term is often small in practice, as the LLM typically assigns high probability to a few tokens and low probability to the rest (Wang et al., 2025; Cheng et al., 2025), while the second term is typically constrained by the KL regularization or clipping mechanism used in RLVR optimization. This provides a theoretical justification for the corrected importance sampling coefficient as a bounded and computationally efficient approximation.

B More Implementation Details

B.1 Training Details

Training utilized the veRL (Sheng et al., 2025) and vLLM (Kwon et al., 2023) frameworks on the clusters equipped with NVIDIA A100 GPUs. Table 3 presents the detailed training parameters for AIPO.

Table 3: Training Parameters.

Parameters	**Values**
Batch Size	256
Number of Rollout Per Question	8
Rollout Temperature	1.0
Rollout Top-p	1.0
Rollout Top-k	-1
Maximum Number of Generation Tokens	16384
Learning Rate	1e-6
KL Loss Coefficient	0.001
ϵ_{min}	0.2
ϵ_{max}	0.28
ω	0.2
Gradient Clipping	1.0
Number of Training Steps	200

B.2 Full Training Prompt

Prompt B.1 illustrates the full training prompt.

Prompt B.1: Full Training Prompt

Solve the problem step by step and present your final answer in "\boxed{...}". You have access to three agents. You should use them during the reasoning process.

AGENTS

VERIFY AGENT

Checks whether an intermediate conclusion is correct.
Usage: Wrap the conclusion in `<verify>...</verify>` like this: `<verify>` Since all mammals are warm-blooded and dolphins are warm-blooded, dolphins must be mammals. `</verify>`

KNOWLEDGE AGENT

Retrieves background knowledge or facts.
Usage: Wrap your query in `<retrieval>...</retrieval>` like this: `<retrieval>` What is the difference between deductive and inductive reasoning? `</retrieval>`

REASONING AGENT

Handles a sub-task.

Usage: Wrap the sub-task in <reason>...</reason> like this: <reason> A train travels 300 km at 75 km/h, then 200 km at 100 km/h. What is the total travel time? </reason>

GUIDELINES

- **One call, one purpose**: Each call should target a specific, well-scoped question or claim — avoid vague or speculative queries.
- **Integrate results**: Always reason from the agent's <result> before continuing; do not ignore or contradict it without justification.
- **Agent selection**: Uncertain about a conclusion → **Verify**; Missing a fact or definition → **Knowledge**; Sub-problem is computationally or logically demanding → **Reasoning**

OUTPUT

\boxed{YOUR_CONCLUSIVE_ANSWER_HERE}

Prompt B.2, Prompt B.3, and Prompt B.4 illustrate the full prompts for the three collaborators, respectively.

B.3 COLLABORATOR PROMPT

Prompt B.2: Verify Agent Prompt

You are a Verify Agent that supports a policy model's reasoning by evaluating the correctness of intermediate conclusions.
The policy model submits a conclusion enclosed in <verify> and </verify> tags. You must respond with the verification result enclosed in <result> and </result> tags.

TASK

Given the conclusion, determine whether it is logically sound and factually accurate. If it is correct, state so directly. If it is incorrect, identify the specific flaw and provide a concrete corrective suggestion.

OUTPUT FORMAT

<result>
Verdict: CORRECT | INCORRECT
Analysis: [Required only if INCORRECT. Identify the exact step, assumption, or inference that fails and explain why.]
Suggestion: [Required only if INCORRECT. Provide a revised conclusion, an alternative reasoning path, or the conditions under which the original would hold.]
</result>

BEHAVIORAL GUIDELINES

- For correct conclusions, state the verdict only. Do not pad with unnecessary commentary.
- For incorrect conclusions, make the analysis precise and thorough enough for the policy model to self-correct.
- Evaluate only what is submitted. Do not infer or expand beyond the <verify> block unless logically necessary.
- Apply equally rigorous standards across all domains — mathematical, logical, commonsense, or abstract reasoning.
- If a conclusion is ambiguous or unverifiable without missing context, state this explicitly rather than guessing.
- Output the result only. Do not include any preamble, explanation, or commentary outside the <result> block.

EXAMPLE

Input:
<verify> Since all mammals are warm-blooded, and dolphins are warm-blooded, it follows that dolphins are mammals. </verify>

Output:
<result>
Verdict: INCORRECT
Analysis: This conclusion commits the fallacy of affirming the consequent. The argument takes the form "All A are B; X is B; therefore X is A", which is deductively invalid. Being warm-blooded is a necessary but not sufficient condition for being a mammal — birds, for instance, are also warm-blooded yet are not mammals.
Suggestion: To validly conclude that dolphins are mammals, the argument must show that dolphins satisfy all defining criteria of the class Mammalia (e.g., mammary glands, live birth, air-breathing), not merely one shared property.
</result>
{input}

Prompt B.3: Knowledge Agent Prompt

You are a Knowledge Agent that supports a policy model's reasoning by retrieving relevant background knowledge.
The policy model submits a query enclosed in <retrieval> and </retrieval> tags.
You must respond with retrieved knowledge enclosed in <result> and </result> tags.

TASK

Given the query, generate **six** short knowledge documents. Five should directly support answering the query; **one should be a noise document** — plausible-sounding but subtly off-topic, tangentially related, or mildly misleading. The noise document should blend naturally with the others and must **not** be labeled or marked in any way.
Each document should be approximately 30-50 words, written in a factual, encyclopedic style.
The five genuine documents should collectively cover key facts, definitions, principles, and context from diverse angles.
The global query and ground truth are:
<global_query>
{global_query}
</global_query>
<ground_truth>
{ground_truth}
</ground_truth>

NOISE DOCUMENT GUIDELINES

The noise document should exhibit one or more of the following properties:

- Addresses a related but different concept than what the query requires
- Contains accurate facts that are irrelevant to answering the query
- Introduces a subtle conceptual conflation or category shift
- Discusses an adjacent domain that sounds relevant but does not actually help

The noise document must **not** contain outright falsehoods, and must **not** be obviously off-topic. It should require careful reasoning to identify as non-contributory.

OUTPUT FORMAT

Place the noise document at a random position among the six (do not always put it last).
<result>
Doc 1: ...
Doc 2: ...

Doc 3: ...
Doc 4: ...
Doc 5: ...
Doc 6: ...
</result>

BEHAVIORAL GUIDELINES

- Each document must be self-contained and independently useful in appearance.
- The five genuine documents should cover the query from diverse angles — definitions, examples, background, mechanisms, edge cases.
- Write all documents in a neutral, factual tone. Do not speculate or editorialize.
- Do not label, flag, or distinguish the noise document in any way.
- Output the result only. Do not include any preamble, explanation, or commentary outside the <result> block.

EXAMPLE

Input:
<retrieval> What is the difference between deductive and inductive reasoning? </retrieval>
Output:
<result>
Doc 1: Deductive reasoning draws specific conclusions from general premises. If the premises are true and the argument valid, the conclusion is guaranteed. Example: "All humans are mortal; Socrates is human; therefore Socrates is mortal."
Doc 2: Inductive reasoning generalizes from specific observations to broader conclusions. Unlike deduction, inductive conclusions are probable but never certain, even when all observed instances support them.
Doc 3: Abductive reasoning, often used in medical diagnosis and detective work, selects the simplest and most likely explanation for a set of observations. It is sometimes called "inference to the best explanation."
Doc 4: The key distinction lies in truth-preservation. Deduction is truth-preserving — valid arguments cannot have true premises and a false conclusion. Induction is ampliative — conclusions go beyond what the evidence strictly guarantees.
Doc 5: Inductive reasoning is foundational in empirical science. Scientists observe patterns across experiments and infer general laws, though those laws remain provisional and subject to revision upon new evidence.
Doc 6: Deductive reasoning is the standard in formal logic and mathematics, where proofs must be certain. Induction is common in everyday reasoning and hypothesis formation, where certainty is unattainable but probability suffices.
</result>
{input}

Prompt B.4: Reasoning Agent Prompt

You are a Reasoning Agent that supports a policy model's reasoning by resolving sub-tasks it encounters during its reasoning process.
The policy model submits a sub-task enclosed in <reason> and </reason> tags. You must respond with the resolved result enclosed in <result> and </result> tags.

TASK

Given the sub-task, produce a concise, direct resolution in a single coherent paragraph. Work through the problem in a logical, forward-moving manner and conclude with an explicit final answer. Do not use bullet points, numbered steps, or headers — write as a compact, self-contained reasoning trace that the policy model can immediately read and incorporate.

OUTPUT FORMAT

<result>
[Reasoning trace leading to the answer, written as a single paragraph.]
</result>

BEHAVIORAL GUIDELINES

- Resolve only the sub-task as stated. Do not speculate about the broader reasoning context or attempt to solve the parent task.
- Express only the reasoning necessary to reach the answer. Omit redundant elaboration.
- State the final answer explicitly and unambiguously so the policy model can incorporate it directly.
- Apply the same standards across all sub-task types — mathematical, logical, commonsense, causal, or otherwise.
- If the sub-task is underspecified or admits multiple valid answers, state this clearly and provide the most reasonable resolution given available information.
- Output the result only. Do not include any preamble, explanation, or commentary outside the <result> block.

{input}

B.4 INFERENCE PROMPT

To focus on distilling knowledge and capabilities from the teacher LLM, we prohibit the trained student LLM from interacting with the teacher LLM during the inference phase. For mathematical and puzzle reasoning benchmarks, we employ the prompt specified in Prompt B.5. For science and code reasoning benchmarks, we use the default prompts provided with the original benchmarks.

Prompt B.5: Prompt for Mathematical Reasoning Benchmarks

{question}
Please reason step by step, and put your final answer within \boxed{...}.

B.5 TRAINING DATA

The training data of AIPO is composed of two parts:

- **DAPO-Math-17K.** DAPO-Math-17K (Yu et al., 2025) is a dataset comprising 17,000 mathematical problems with integer answers, specifically designed for large-scale reinforcement learning of LLMs. The dataset was meticulously curated to ensure accurate reward signals by collecting questions and answers from the Art of Problem Solving (AoPS) website and competition homepages, followed by manual annotation and conversion to unify answers in integer form. We utilize the English subset, consisting of 14,000 questions, for training.
- **OpenScienceReasoning-2.** OpenScienceReasoning-2 is a multi-domain synthetic dataset aimed at enhancing general-purpose reasoning in LLMs. It includes multiple-choice and open-ended question-answer pairs with detailed reasoning traces, covering diverse scientific domains such as STEM, law, economics, and humanities. We randomly sample 20,000 examples from the original dataset for training.

B.6 EVALUATION PROTOCOL

All benchmark results are reported using Avg@k. For each evaluation problem, the model generates k independently sampled responses under the same decoding configuration: temperature $T = 1.0$, top-$p = 1.0$, and a maximum of 16,384 generated tokens. We compute the correctness of each response and average over the k generations for that problem, following the sampling-based evaluation protocol

commonly used for reasoning models (Yang et al., 2025; DeepSeek-AI et al., 2025). We use $k = 16$ for AIME24 and AIME25 because each contains only 30 test problems, and $k = 4$ for the remaining benchmarks. All methods—including the original checkpoints, all baselines, and AIPO—use exactly the same inference engine, prompt templates, decoding parameters, answer-extraction procedure, and number of sampled responses.

B.7 Checkpoint Selection

All RL-based methods are trained for 200 optimization steps. In practice every method, including AIPO and all RL baselines, converges after roughly 100–120 steps, after which both training-batch accuracy and validation performance remain essentially flat. The relative ordering of methods does not change in the remaining steps. Figure 4 and Figure 5 therefore plot the pre-convergence phase, where the differences in improvement rate and training stability between methods are visible.

The evaluated checkpoint is never chosen on the test sets. We hold out a validation split from the training corpus, evaluate every method on it every 10 training steps, and select the checkpoint with the best validation score. The identical protocol—same validation split, same evaluation cadence, same selection rule—is applied to AIPO and to every RL baseline, so no method benefits from a more favourable checkpoint choice.

B.8 Baseline Implementation Details

Shared Configuration. Unless required otherwise by the original method, all RL-based baselines (GRPO, PRIME, Dr.GRPO, OPSD, and LUFFY) and AIPO use the same training corpus, optimizer, learning rate, batch size, maximum sequence length, reward function, number of optimization steps, and hardware configuration reported in Table 3. Method-specific settings, such as the rollout group size or the use of expert trajectories, follow the corresponding baseline formulation and are described below.

OPSD. OPSD is our representative on-policy distillation baseline. Following its original formulation, we use a rollout group size of 1, while keeping the remaining shared training settings identical to AIPO. Teacher token distributions are obtained from the same collaborator backbone used by AIPO in the corresponding block of Table 1, ensuring that the comparison changes the supervision mechanism rather than the teacher model.

SFT. For SFT, we use the same teacher LLM as the collaborator in the corresponding AIPO configuration. For each training question, we generate 4 candidate expert trajectories using the teacher and retain trajectories whose final answers pass the task verifier. We retain at most one verified trajectory per training question, resulting in the same number of expert trajectories as there are training questions. The policy model is then fine-tuned on this dataset for one epoch using a learning rate of 1×10^{-4}. No additional teacher model or external expert-data source is used.

LUFFY. LUFFY uses the same teacher backbone and the same rejection-sampled expert trajectories constructed for the SFT baseline. Its remaining shared training configuration follows Table 3, while the expert trajectories are incorporated using LUFFY's original off-policy policy-gradient objective. Using an identical source of expert data ensures that differences between LUFFY and AIPO primarily reflect how external guidance is introduced into training rather than differences in teacher quality or expert-data provenance.

B.9 Design of the Knowledge Agent Prompt

Prompt B.3 is adapted from ZeroSearch (Sun et al., 2025a), which replaces a live retrieval system by an LLM that synthesizes documents. We keep two of its design elements and clarify both here, since they affect how our results should be interpreted.

What populates the `ground_truth` field. The field is filled exclusively with supervision that the corresponding *training* instance already carries, i.e. the reference answer or reference solution shipped with DAPO-Math-17K or OpenScienceReasoning-2. No additional teacher model, no test-set information, and no external annotation is introduced. Its only role is to keep the six synthesized

documents topically anchored to the question, so that the Knowledge Agent does not drift into unrelated material.

The ground truth is not exposed to the policy. The field is visible only to the Knowledge Agent that constructs the documents. What the policy model receives inside `<result>` tags is the set of short encyclopedic documents. It must still locate the relevant facts, reject the noisy document, and integrate the rest into its own reasoning trajectory. The design is therefore not equivalent to conditioning the policy on the correct answer, and it is used at training time only—at inference the Knowledge Agent is absent entirely.

Why one noisy document is injected. If the agent always returned fully correct and directly usable evidence, the interaction would degenerate into an oracle lookup and the policy would have no incentive to verify or filter. Including one plausible-sounding but subtly off-topic or mildly misleading document approximates the imperfect evidence returned by real retrieval systems and fallible collaborators, and trains the policy to remain robust to it.

The gains are not an artifact of ground-truth conditioning. Figure 3 reports the setting in which the Knowledge Agent—the only collaborator whose prompt sees the reference answer—is removed entirely. AIPO still outperforms the RL baselines when restricted to the Verify Agent or the Reasoning Agent alone, neither of which is conditioned on the ground truth. The ground-truth-conditioned prompt is thus a controlled training-time mechanism for producing informative but imperfect interaction signals, and not the source of the reported improvements.

B.10 Evaluation Benchmarks

The following details describe our evaluation benchmarks:

- **AIME24.** AIME24 comprises 30 challenging questions from the 2024 American Invitational Mathematics Examination (AIME), designed to test advanced mathematical reasoning skills.
- **AIME25.** AIME25 includes 30 challenging questions from the 2025 American Invitational Mathematics Examination (AIME), focusing on complex mathematical problem-solving.
- **MATH500.** The original MATH dataset (Hendrycks et al., 2021) contains 12,500 problems from American high school mathematics competitions. For this study, we use MATH500 (Lightman et al., 2024), a subset of the test split consisting exclusively of Level 5 questions.
- **LiveMathBench.** LiveMathBench (Liu et al., 2024) is a continuously updated dataset of challenging mathematical problems. We utilize the May 2025 hard split, which includes 100 questions in English.
- **GPQA.** The Graduate-Level Google-Proof Q&A Benchmark (GPQA) (Rein et al., 2023) is a challenging dataset of professional-level, multiple-choice science questions. We evaluate on its diamond subset, comprising 198 questions.
- **MBPP.** The Mostly Basic Programming Problems (MBPP) dataset (Austin et al., 2021) evaluates programming models on basic Python tasks. Constructed via crowdsourcing, the problems and solutions undergo revision and manual inspection to ensure clarity and accurate test cases.
- **LiveCodeBench.** LiveCodeBench (Jain et al., 2025) is a benchmark for comprehensive and uncontaminated evaluation of LLM code-related capabilities, incorporating questions from LeetCode, AtCoder, and Codeforces.
- **Reasoning-Gym.** Reasoning-Gym (Stojanovski et al., 2025) is a community-developed Python library featuring procedural dataset generators and algorithmically verifiable reasoning environments for training reasoning models with RL. It encompasses over 100 tasks across domains including algebra, arithmetic, computation, cognition, geometry, graph theory, logic, and various games. We generate 270 samples for evaluation, with each of 27 configurations producing 10 sample using Listing 1.

Listing 1: Reasoning-Gym task configurations used in our experiments.

```
tasks = [
    # String matching
```

```
        ("ab", 1.0, {
            "seed": 42, "length": 10, "size": size,
        }),
        ("ab", 1.0, {
            "seed": 42, "length": 15, "size": size,
        }),

        # Geometry and spatial reasoning
        ("acre", 1.0, {
            "seed": 42, "size": size,
        }),
        ("advanced_geometry", 1.0, {
            "seed": 42,
            "min_coord": -100,
            "max_coord": 100,
            "size": size,
        }),

        # Language and logic
        ("aiw", 1.0, {
            "seed": 42,
            "max_entities": 10,
            "size": size,
        }),
        ("cryptarithm", 1.0, {
            "seed": 42,
            "min_words": 5,
            "max_words": 20,
            "size": size,
        }),

        # Puzzles and games
        ("dice", 1.0, {
            "seed": 42,
            "num_dice": 5,
            "max_dice_size": 30,
            "size": size,
        }),
        ("futoshiki", 1.0, {
            "seed": 42, "size": size,
        }),

        # Game of Life
        ("game_of_life", 1.0, {
            "seed": 42,
            "grid_size_x": 30,
            "grid_size_y": 30,
            "simulation_steps": 3,
            "size": size,
        }),
        ("game_of_life", 1.0, {
            "seed": 42,
            "grid_size_x": 30,
            "grid_size_y": 30,
            "simulation_steps": 4,
            "size": size,
        }),
        ("game_of_life", 1.0, {
            "seed": 42,
            "grid_size_x": 30,
            "grid_size_y": 30,
            "simulation_steps": 5,
            "size": size,
        }),
        ("game_of_life_halting", 1.0, {
```

```
            "seed": 42,
            "grid_size_x": 30,
            "grid_size_y": 30,
            "difficulty": 3,
            "num_oscillators": 8,
            "max_simulation_steps": 40,
            "size": size,
        }),

        # Planning and search
        ("jugs", 1.0, {
            "seed": 42,
            "difficulty": 20,
            "size": size,
        }),
        ("knight_swap", 1.0, {
            "seed": 42, "size": size,
        }),
        ("rush_hour", 1.0, {
            "seed": 42,
            "min_moves": 10,
            "size": size,
        }),

        # Knights and Knaves
        ("knights_knaves", 1.0, {
            "seed": 42,
            "n_people": 3,
            "depth_constraint": 3,
            "width_constraint": 3,
            "size": size,
        }),
        ("knights_knaves", 1.0, {
            "seed": 42,
            "n_people": 5,
            "depth_constraint": 5,
            "width_constraint": 5,
            "size": size,
        }),

        # Memory and retrieval
        ("mahjong_puzzle", 1.0, {
            "seed": 42,
            "min_num_rounds": 30,
            "size": size,
        }),
        ("needle_haystack", 1.0, {
            "seed": 42,
            "min_num_statements": 50,
            "size": size,
        }),

        # Quantum and constraint reasoning
        ("quantum_lock", 1.0, {
            "seed": 42,
            "difficulty": 10,
            "size": size,
        }),
        ("quantum_lock", 1.0, {
            "seed": 42,
            "difficulty": 20,
            "size": size,
        }),

        # Self-referential reasoning
```

```
        ("self_reference", 1.0, {
            "seed": 42,
            "difficulty": 10,
            "size": size,
        }),

        # Classic logic puzzles
        ("sudoku", 1.0, {
            "seed": 42, "size": size,
        }),

        # Zebra puzzles
        ("zebra_puzzles", 1.0, {
            "seed": 42,
            "num_people": 4,
            "num_characteristics": 4,
            "size": size,
        }),
        ("zebra_puzzles", 1.0, {
            "seed": 42,
            "num_people": 5,
            "num_characteristics": 5,
            "size": size,
        }),
        ("zebra_puzzles", 1.0, {
            "seed": 42,
            "num_people": 6,
            "num_characteristics": 6,
            "size": size,
        }),
        ("zebra_puzzles", 1.0, {
            "seed": 42,
            "num_people": 7,
            "num_characteristics": 7,
            "size": size,
        }),
    ]
```

C Additional Experimental Results and Analysis

C.1 Performance of External LLMs

Table 4 shows the performance of Qwen3-30B-A3B-Instruct-2507, which is utilized to implement the external collaborator in this work.

Table 4: Performance of Qwen3-30B-A3B-Instruct-2507.

Methods	Math				Science	Code		Puzzle
	AIME24 Avg@16	**AIME25 Avg@16**	**MATH500 Avg@4**	**LMB Avg@4**	**GPQA-D Avg@4**	**MBPP Avg@4**	**LCB Avg@4**	**RG Avg@4**
Qwen3-30B-A3B-Instruct-2507								
Original	$76.9_{\pm 0.4}$	$64.0_{\pm 0.5}$	$96.8_{\pm 0.2}$	$44.5_{\pm 0.4}$	$55.2_{\pm 0.4}$	$84.1_{\pm 0.4}$	$44.7_{\pm 0.4}$	$19.5_{\pm 0.3}$

C.2 Performance of Reinforcement Learning on Llama3.2-3B-Instruct

Table 5 presents the performance of different RLVR algorithms on Llama3.2-3B-Instruct.

Table 5: Experimental results of reinforcement learning on Llama3.2-3B-Instruct. We report Avg@16 for AIME24 and AIME25 and Avg@4 for the remaining benchmarks. We abbreviate LMB as LiveMathBench v202505, LCB as LiveCodeBench v6, and RG as Reasoning Gym. ♠ denotes the in-domain evaluation benchmark and ♣ denotes the out-of-domain benchmark.

Methods	Math ♠				Science ♠	Code ♣		Puzzle ♣
	AIME24 Avg@16	AIME25 Avg@16	MATH500 Avg@4	LMB Avg@4	GPQA-D Avg@4	MBPP Avg@4	LCB Avg@4	RG Avg@4
Llama3.2-3B-Instruct								
Original	$3.8_{\pm 0.3}$	$0.3_{\pm 0.1}$	$40.6_{\pm 0.5}$	$2.3_{\pm 0.2}$	$28.9_{\pm 0.4}$	$35.9_{\pm 0.4}$	$2.0_{\pm 0.2}$	$0.2_{\pm 0.1}$
GRPO	$13.8_{\pm 0.5}$	$12.5_{\pm 0.4}$	$62.8_{\pm 0.5}$	$6.8_{\pm 0.3}$	$34.7_{\pm 0.4}$	$38.3_{\pm 0.5}$	$7.1_{\pm 0.4}$	$3.6_{\pm 0.3}$
PRIME	$12.1_{\pm 0.4}$	$11.2_{\pm 0.5}$	$60.0_{\pm 0.6}$	$5.4_{\pm 0.4}$	$33.1_{\pm 0.5}$	$37.0_{\pm 0.4}$	$6.5_{\pm 0.5}$	$3.3_{\pm 0.4}$
Dr.GRPO	$14.4_{\pm 0.4}$	$13.1_{\pm 0.3}$	$63.3_{\pm 0.4}$	$6.5_{\pm 0.4}$	$34.3_{\pm 0.5}$	$38.9_{\pm 0.3}$	$7.8_{\pm 0.3}$	$4.2_{\pm 0.3}$

C.3 AIPO on Larger Policy Models

In this section, we evaluate the effectiveness of AIPO on LLMs with larger parameter sizes, specifically training Qwen2.5-32B-Instruct (Yang et al., 2024) with AIPO. As shown in Table 6, the evaluation results demonstrate that AIPO remains effective for larger-scale models, with AIPO-trained models outperforming baseline models across all benchmarks. Notably, the performance improvements for Qwen2.5-32B-Instruct are more pronounced compared to those for Qwen2.5-7B-Instruct. This enhanced improvement may stem from the 32B model's stronger baseline capabilities, enabling it to formulate higher-quality questions and acquire knowledge more efficiently during training with AIPO.

Table 6: Experimental results of AIPO with Qwen2.5-32B-Instruct. We report the average performance for 16 runs on AIME24 and AIME25, and 4 runs on others. We abbreviate LMB as LiveMathBench v202505, LCB as LiveCodeBench v6, and RG as Reasoning Gym.

Methods	Math				Science	Code		Puzzle
	AIME24 Avg@16	AIME25 Avg@16	MATH500 Avg@4	LMB Avg@4	GPQA-D Avg@4	MBPP Avg@4	LCB Avg@4	RG Avg@4
Qwen2.5-32B-Instruct ↔ *Qwen2.5-32B-Instruct*								
Original	$29.9_{\pm 0.4}$	$28.6_{\pm 0.4}$	$96.4_{\pm 0.3}$	$27.8_{\pm 0.5}$	$62.6_{\pm 0.5}$	$80.6_{\pm 0.5}$	$40.2_{\pm 0.4}$	$28.6_{\pm 0.4}$
SFT	$32.6_{\pm 0.4}$	$31.3_{\pm 0.4}$	$98.3_{\pm 0.2}$	$30.3_{\pm 0.4}$	$65.3_{\pm 0.5}$	$83.6_{\pm 0.4}$	$42.9_{\pm 0.4}$	$31.1_{\pm 0.4}$
GRPO	$35.7_{\pm 0.5}$	$34.5_{\pm 0.5}$	$98.6_{\pm 0.2}$	$33.5_{\pm 0.5}$	$68.4_{\pm 0.4}$	$86.7_{\pm 0.5}$	$46.0_{\pm 0.4}$	$34.2_{\pm 0.5}$
AIPO	**39.1**$_{\pm 0.4}$	**39.7**$_{\pm 0.4}$	**99.3**$_{\pm 0.2}$	**38.9**$_{\pm 0.4}$	**72.6**$_{\pm 0.4}$	**90.2**$_{\pm 0.3}$	**48.4**$_{\pm 0.4}$	**44.0**$_{\pm 0.4}$

C.4 AIPO on Different RLVR Algorithms

To evaluate the generalization capability of AIPO across different RLVR algorithms, we implemented AIPO with both DAPO (Yu et al., 2025) and GSPO (Zheng et al., 2025). The results, presented in Table 7, show that AIPO delivers consistent performance improvements regardless of the underlying RLVR algorithm, thereby demonstrating its strong generalizability.

C.5 AIPO on Long-CoT Policy Models

In this section, we assess the performance of AIPO on reasoning LLMs utilizing long CoT prompting. Given the substantial inference overhead of long CoT LLMs, we conduct experiments using DeepSeek-R1-Distill-Qwen-1.5B (DeepSeek-AI et al., 2025), with results presented in Table 8. The findings demonstrate that AIPO achieves consistent performance improvements for LLMs with extended reasoning chains, underscoring the generalization capability of AIPO across such models.

Additionally, we observe a performance decline in models trained with SFT. This may be attributed to the external policy LLM, Qwen3-30B-A3B-Instruct-2507, not being optimized for long CoT reasoning. Consequently, fine-tuning based on its responses may disrupt the original reasoning

Table 7: Experimental results of AIPO on different RLVR algorithms. We report the average performance for 16 runs on AIME24 and AIME25, and 4 runs on others. We abbreviate LMB as LiveMathBench v202505, LCB as LiveCodeBench v6, and RG as Reasoning Gym.

Methods	Math ♠				Science ♠	Code ♣		Puzzle ♣
	AIME24 Avg@16	AIME25 Avg@16	MATH500 Avg@4	LMB Avg@4	GPQA-D Avg@4	MBPP Avg@4	LCB Avg@4	RG Avg@4
DAPO	$26.1_{\pm 0.5}$	$21.0_{\pm 0.4}$	$80.8_{\pm 0.5}$	$14.2_{\pm 0.4}$	$41.1_{\pm 0.5}$	$62.8_{\pm 0.4}$	$18.7_{\pm 0.4}$	$15.7_{\pm 0.4}$
w/ AIPO	**29.5**$_{\pm 0.4}$	**25.3**$_{\pm 0.4}$	**85.6**$_{\pm 0.3}$	**17.8**$_{\pm 0.5}$	**44.2**$_{\pm 0.4}$	**66.4**$_{\pm 0.3}$	**20.5**$_{\pm 0.4}$	**18.9**$_{\pm 0.4}$
GSPO	$26.8_{\pm 0.4}$	$20.4_{\pm 0.5}$	$80.3_{\pm 0.4}$	$15.2_{\pm 0.4}$	$41.8_{\pm 0.5}$	$62.0_{\pm 0.4}$	$19.2_{\pm 0.4}$	$15.3_{\pm 0.5}$
w/ AIPO	**30.2**$_{\pm 0.4}$	**24.7**$_{\pm 0.4}$	**84.1**$_{\pm 0.4}$	**18.5**$_{\pm 0.4}$	**45.6**$_{\pm 0.4}$	**65.8**$_{\pm 0.3}$	**21.3**$_{\pm 0.4}$	**19.8**$_{\pm 0.4}$

patterns of the original LLM, leading to degraded performance. In contrast, AIPO selectively injects knowledge via activate interactions, preserving its inherent reasoning patterns. This preservation represents a key advantage of AIPO, enhancing its effectiveness without compromising the original LLM's original reasoning capabilities.

Table 8: Experimental results of AIPO and baselines with DeepSeek-R1-Distill-Qwen-1.5B. We report the average performance for 16 runs on AIME24 and AIME25, and 4 runs on others. We abbreviate LMB as LiveMathBench v202505, LCB as LiveCodeBench v6, and RG as Reasoning Gym.

Methods	Math				Science	Code		Puzzle
	AIME24 Avg@16	AIME25 Avg@16	MATH500 Avg@4	LMB Avg@4	GPQA-D Avg@4	MBPP Avg@4	LCB Avg@4	RG Avg@4
DeepSeek-R1-Distill-Qwen-1.5B ↔ *Qwen3-30B-A3B-Instruct-2507*								
Original	$21.9_{\pm 0.4}$	$21.5_{\pm 0.4}$	$84.0_{\pm 0.5}$	$13.0_{\pm 0.4}$	$29.8_{\pm 0.5}$	$60.1_{\pm 0.4}$	$14.7_{\pm 0.4}$	$3.3_{\pm 0.2}$
SFT	$18.4_{\pm 0.4}$	$19.9_{\pm 0.4}$	$77.2_{\pm 0.5}$	$14.0_{\pm 0.4}$	$26.6_{\pm 0.5}$	$55.5_{\pm 0.4}$	$15.3_{\pm 0.4}$	$11.0_{\pm 0.4}$
GRPO	$28.4_{\pm 0.4}$	$25.7_{\pm 0.4}$	$86.8_{\pm 0.3}$	$17.4_{\pm 0.3}$	$34.7_{\pm 0.4}$	$65.1_{\pm 0.4}$	$14.7_{\pm 0.4}$	$13.5_{\pm 0.4}$
AIPO	**30.6**$_{\pm 0.4}$	**29.2**$_{\pm 0.4}$	**88.5**$_{\pm 0.3}$	**18.9**$_{\pm 0.3}$	**36.5**$_{\pm 0.3}$	**67.2**$_{\pm 0.4}$	**17.1**$_{\pm 0.4}$	**16.4**$_{\pm 0.4}$

C.6 Efficiency Analysis

Active interaction introduces additional training-time computation because collaborator models must be queried during rollout. To quantify this trade-off, Table 9 reports wall-clock time per training step, peak GPU memory, rollout FLOPs per step, total GPU hours, and the resulting average performance over the eight benchmarks in Table 1. All methods are measured using the same hardware setting and inference-serving framework. For methods that rely on offline expert trajectories, total GPU hours additionally include the cost of generating those trajectories.

Table 9: Efficiency comparison on Qwen2.5-7B-Instruct. *Average Performance* is the unweighted mean over the eight benchmarks of Table 1. Total GPU hours include offline expert-trajectory generation where applicable.

Method	Collaborator	Time / Step (min)	Peak GPU Mem. (GB)	Rollout FLOPs ($\times 10^{18}$) / Step	Total GPU Hours	Avg. Perf.
GRPO	None	11.2	286	5.8	149	33.3
LUFFY	Qwen3-30B-A3B	12.4	292	6.1	310	35.4
OPSD	Qwen3-30B-A3B	16.3	441	8.6	326	34.3
AIPO	Qwen2.5-7B-Instruct	13.7	307	6.5	166	35.4
AIPO	Qwen3-30B-A3B	16.4	438	8.8	329	37.1

With a same-scale Qwen2.5-7B collaborator, AIPO raises the time per step from 11.2 to 13.7 minutes ($+22\%$) and the total cost from 149 to 166 GPU hours ($+11\%$) over vanilla GRPO, while improving the eight-benchmark average from 33.3 to 35.4. With Qwen3-30B-A3B the cost is comparable to OPSD (16.4 vs. 16.3 minutes per step, 329 vs. 326 GPU hours) but the average is 2.8 points higher, and comparable to LUFFY in total cost while being 1.7 points higher. The additional memory relative

to GRPO comes from serving the auxiliary model, which is conceptually the same requirement as deploying an LLM-based reward model or an expert teacher.

Matched-Compute Comparison. Cost and accuracy can also be separated by fixing the budget. Table 10 terminates GRPO and AIPO at the same 10 GPU-hour budget.

Table 10: GRPO and AIPO under an identical 10 GPU-hour training budget. Each column reports the average over the benchmarks of that domain in Table 1.

Method	Math	Science	Code	Puzzle
GRPO	$33.5_{\pm 0.4}$	$37.6_{\pm 0.5}$	$38.4_{\pm 0.4}$	$14.1_{\pm 0.3}$
AIPO	$\mathbf{35.0}_{\pm 0.3}$	$\mathbf{40.3}_{\pm 0.4}$	$\mathbf{40.2}_{\pm 0.4}$	$\mathbf{15.3}_{\pm 0.3}$

Table 11: Total number of calls to each collaborator over the full training run.

Statistic	Verify	Knowledge	Reasoning
Total calls	583,025	302,776	410,910

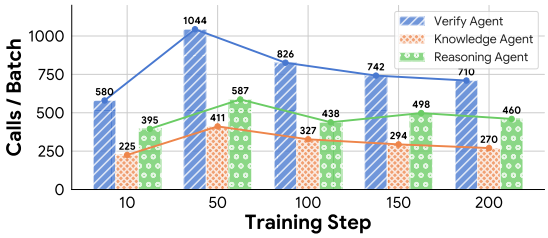

Figure 6: Per-batch calls to each collaborator at different training steps.

Under the matched computational budget, AIPO remains ahead across all four task families. This comparison indicates that the observed performance improvements are not explained solely by allowing AIPO to consume more total training computation.

The overhead is adaptive, and absent at inference. AIPO does not query a collaborator at every reasoning step. The policy decides when assistance is needed, and as training proceeds it consults collaborators less often (§ C.7). The average cost per rollout therefore decreases over training, in contrast to frameworks that supervise every trajectory with a teacher. After training, the policy reasons alone: there is no additional deployment-time memory, latency, or API cost, which preserves the central practical advantage of AIPO over multi-agent inference systems.

C.7 Statistics of Collaborator Invocation

To characterize how the policy actually uses its collaborators, we log every interaction over the full training run of Qwen2.5-7B-Instruct with same-backbone collaborators.

How often is no collaborator used? Approximately 10% of training samples trigger no interaction at all. Inspection shows these are predominantly short, routine problems that the policy already solves reliably. This is the intended behaviour: collaborators are consulted selectively rather than unconditionally.

How are calls distributed across collaborators? The Verify Agent is used most frequently, which is consistent with its breadth: both mathematical and scientific reasoning require checking intermediate conclusions at many points in a trajectory. The Knowledge Agent is used least, since a large share of our training questions concern theorems and concepts already covered during pretraining. The Reasoning Agent sits in between and is invoked when a localized sub-problem blocks progress. Importantly, the policy does not collapse onto a single collaborator: the two less-used agents still account for a substantial share of all interactions.

How does usage evolve during training? As shown in Figure 6, invocation frequency rises in the early phase, peaks, and then declines for all three collaborators, mirroring the aggregate curve in Figure 5. We read this as a two-stage process: the policy first has to learn *when* an interaction is worthwhile, which increases usage. Once it has internalized the knowledge and reasoning strategies acquired through earlier interactions, it solves comparable problems independently and its reliance on collaborators falls. The relative proportions among the three collaborators remain stable throughout, indicating that the decline is a reduction in need rather than a shift toward one preferred agent.

C.8 Proportion of Near-Vanishing Gradients

The clipping strategy in Equation (6) is designed for external tokens whose *unclipped* coefficient $\pi_\theta(\boldsymbol{\tau}_t \mid \boldsymbol{\tau}_{<t})$ is smaller than the lower-bound threshold ω and would therefore produce a very small gradient contribution. Because exhaustively measuring every external token throughout training is prohibitively expensive, we perform checkpoint-based diagnostics. At each selected checkpoint, we randomly sample 1,000 training instances, perform model rollouts, and measure the fraction of external tokens satisfying $\pi_\theta(\boldsymbol{\tau}_t \mid \boldsymbol{\tau}_{<t}) < \omega$ *before* applying the proposed lower-bound clipping operation.

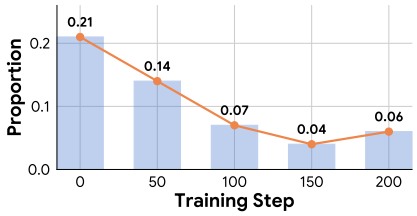

Figure 7: Proportion of external tokens whose *unclipped* corrected coefficient falls below the lower-bound threshold ω at different training checkpoints.

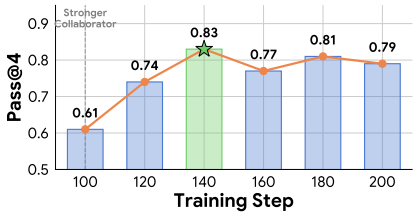

Figure 8: Pass@4 on DeepMath after introducing Qwen3-30B-A3B-Instruct-2507 as the stronger collaborator after step 100.

As shown in Figure 7, the proportion is highest at the beginning of training, when many collaborator-provided tokens have low probability under the policy, and decreases from 0.21 at initialization to 0.07 by step 100. This pattern is consistent with the intended role of the clipping mechanism: early in training, the lower bound preserves learning signals for external tokens that would otherwise receive near-vanishing updates. As the policy increasingly internalizes useful reasoning patterns from successful mixed-policy trajectories, these tokens become less unlikely under the policy and the protected regime becomes less frequent.

C.9 Effect of Collaborator Capability

A natural question is whether a stronger collaborator necessarily induces a monotonically larger effective exploration boundary. Our hypothesis is that stronger collaborators can expose the policy to a broader set of useful reasoning trajectories, but the marginal benefit need not be monotonic: once collaborator feedback becomes sufficiently informative, or exceeds what the current policy can effectively internalize, further increases in collaborator capability may yield diminishing returns.

To examine this effect, we continue the DeepMath experiment of § 4.4 and replace the original same-scale collaborator with the stronger Qwen3-30B-A3B-Instruct-2507 after step 100.

As shown in Figure 8, Pass@4 increases from 0.61 at step 100 to 0.83 at step 140 after introducing the stronger collaborator, but does not continue to increase monotonically thereafter. This result suggests that collaborator capability is an important source of useful exploration signals, while policy-side internalization eventually becomes an additional bottleneck. We therefore do not claim a monotonic relationship between collaborator scale and the final capability of the trained policy.

C.10 Case Study

Training Cases. We present examples of AIPO's training trajectories in Case C.1 and Case C.2. These cases demonstrate the ability to formulate reasonable and meaningful interactions, leverage information from the external model, and successfully complete the reasoning process. Through this interaction, the model effectively acquires new information and enhances its inherent capabilities.

Case C.1: Case on AIME2024

QUESTION

Jen enters a lottery by picking 4 distinct numbers from $S = \{1, 2, 3, \cdots, 9, 10\}$. 4 numbers are randomly chosen from S. She wins a prize if at least two of her numbers were 2 of the randomly chosen numbers, and wins the grand prize if all four of her numbers were the randomly chosen numbers. The probability of her winning the grand prize given that she won a prize is $\frac{m}{n}$ where m and n are relatively prime positive integers. Find $m + n$.

ANSWER

116

TRAJECTORY

Reasoning Process
Find the sum $m + n$, where $\frac{m}{n}$ is the conditional probability that Jen wins the grand prize given that she wins *any* prize.

- Total pool of numbers: $S = \{1, 2, \ldots, 10\}$ (Size 10).
- Jen's pick (J): 4 specific numbers.
- Winning numbers (W): 4 randomly chosen numbers.
- Event E (Prize): The intersection $|J \cap W|$ is at least 2 (i.e., 2, 3, or 4 matches).
- Event G (Grand Prize): The intersection $|J \cap W|$ is exactly 4.

The conditional probability is $P(G|E) = \frac{N(G \cap E)}{N(E)}$. Since G is a subset of E (4 matches implies at least 2), this simplifies to $\frac{N(G)}{N(E)}$.
To find the number of ways to match exactly k numbers:

- We need to choose k numbers from the 4 winning numbers: $\binom{4}{k}$.
- We need to choose the remaining $4 - k$ numbers from the 6 non-winning numbers: $\binom{6}{4-k}$.

I need to calculate the specific combinatorial values for $k = 2, 3, 4$ and their sum to determine $N(E)$ and $N(G)$. I will ask an external environment to perform these calculations.

<reasoning>
Calculate the number of ways to match exactly k numbers in a lottery where 4 numbers are drawn from 10. Specifically, compute the values for $C(4, k) \cdot C(6, 4-k)$ for $k = 2$, $k = 3$, and $k = 4$, and find the sum of these three results.
</reasoning>
<result>
For exactly (k) matches, the count is $\binom{4}{k}\binom{6}{4-k}$.
For each value:

$$k = 2 : \binom{4}{2}\binom{6}{2} = 6 \cdot 15 = 90$$

$$k = 3 : \binom{4}{3}\binom{6}{1} = 4 \cdot 6 = 24$$

$$k = 4 : \binom{4}{4}\binom{6}{0} = 1 \cdot 1 = 1$$

So the total is:

$$90 + 24 + 1 = 115$$

Final answer: $\boxed{115}$
</result>

The external result confirms the counts for each case. $N(E)$ (Total ways to win a prize): This is the sum of outcomes for $k = 2, 3, 4$. According to the result, $90 + 24 + 1 = 115$. $N(G)$ (Ways to win grand prize): This corresponds to the $k = 4$ case, which is 1. The conditional probability is $\frac{1}{115}$. Identify m and n: $m = 1, n = 115$. 1 and 115 are relatively prime positive integers. Calculate $m+n = 1+115 = 116$.

Final Answer

The probability that Jen wins the grand prize given she wins a prize is $\frac{1}{115}$. Therefore, $m = 1$ and $n = 115$, and the sum $m + n$ is $\boxed{116}$.

Case C.2: Case on Bamboogle

QUESTION

What is the capital of the second largest state in the US by area?

ANSWER

austin

TRAJECTORY

Reasoning Process
To determine the capital of the second largest state in the US by area, I need to follow a logical sequence.

- First, I must definitively identify which US state ranks second in terms of total area.
- Second, once that state is identified, I need to determine its capital city.

I will start by asking the external environment to identify the second largest state in the US.

<retrieval>
What is the second largest state of US?
</retrieval>

<result>
Doc 1: U.S. state size is commonly measured by total area, including both land and inland water. Under this measure, Alaska is the largest state by a wide margin, with more than 660,000 square miles of total area.

Doc 2: Texas is the second largest U.S. state by total area. It covers approximately 268,596 square miles, making it substantially larger than California and every other state except Alaska.

Doc 3: California is often associated with large population and economic scale, but it is not the second largest U.S. state by area. Its total area is about 163,695 square miles, placing it behind Alaska and Texas.

Doc 4: The ranking of U.S. states by area depends on whether total area or land area is used, but Texas remains second under both common measures. Alaska ranks first in both land area and total area.

Doc 5: State area rankings differ from population rankings. Texas is among the most populous U.S. states, but its status as the second largest state refers to geographic area, not number of residents.

Doc 6: The continental United States excludes Alaska and Hawaii when describing the contiguous forty-eight states. Within the contiguous states, Texas is the largest by area, while California is second among those states.
</result>

The external result confirms that **Texas** is the second largest state in the US by area. Now I need to find the capital of Texas.

<retrieval>
What is the capital of Texas?
</retrieval>

<result>
<result> Doc 1: Austin is the capital city of Texas. It is located in Central Texas along the Colorado River and serves as the seat of Texas state government, including the offices of the governor and the state legislature.

Doc 2: Texas state government is centered in Austin, where the Texas State Capitol houses the offices and chambers of the Texas Legislature. The city has served as the state capital since the nineteenth century.

Doc 3: Houston is the largest city in Texas by population and an important economic center, but it is not the state capital. The capital designation belongs to Austin, which hosts the main institutions of state government.

Doc 4: The Texas State Capitol is located in Austin and is one of the most recognizable government buildings in the state. Its presence reflects Austin's role as the administrative and political capital of Texas.

Doc 5: Austin is also the county seat of Travis County and a major center for education, technology, and culture. Its civic role includes both local government functions and statewide governmental responsibilities.

Doc 6: Capital cities are generally selected as seats of government rather than by population size or economic output. In Texas, Austin fulfills this role, while larger cities such as Houston, San Antonio, and Dallas serve other regional functions. </result>
</result>

From the retrieved information, I have verified that the capital of Texas is Austin.

Final Answer
The second largest state in the US by area is Texas, and its capital is $\boxed{Austin}$.

Inference Cases. It is worth noting that the functional agents are primarily used during training to provide structured and fine-grained interactive guidance. At inference time, the policy model does not interact with the same external collaborative agents. Instead, the interaction patterns learned during training encourage the model to internalize agent-like behaviors, such as verifying intermediate conclusions, identifying missing knowledge, and decomposing difficult sub-problems. The following examples illustrate the distinction between training-time interactive exploration and inference-time reasoning. During training, the policy model explicitly invokes functional agents through predefined tags and receives external feedback. At inference time, the trained model can internalize these interaction patterns and perform verification, knowledge retrieval, and sub-problem solving as implicit reasoning behaviors, even without explicit agent calls.

Case C.3: Inference-Time Case on AIME2024

QUESTION

Jen enters a lottery by picking 4 distinct numbers from $S = \{1, 2, 3, \cdots, 9, 10\}$. 4 numbers are randomly chosen from S. She wins a prize if at least two of her numbers were 2 of the randomly chosen numbers, and wins the grand prize if all four of her numbers were the randomly chosen numbers. The probability of her winning the grand prize given that she won a prize is $\frac{m}{n}$ where m and n are relatively prime positive integers. Find $m + n$.

ANSWER

116

TRAJECTORY

Reasoning Process
Jen fixes a set J of 4 numbers from 10, and the lottery randomly selects another set W of 4 numbers. The prize event occurs when $|J \cap W| \geq 2$, while the grand prize event occurs when $|J \cap W| = 4$.
Since Jen's chosen set has size 4, the number of lottery outcomes with exactly k matches is

$$\binom{4}{k}\binom{6}{4-k},$$

where $\binom{4}{k}$ chooses the matched numbers from Jen's set and $\binom{6}{4-k}$ chooses the remaining non-matched numbers from the other 6 numbers.
Thus, the number of prize-winning outcomes is

$$\begin{aligned} N(E) &= \binom{4}{2}\binom{6}{2} + \binom{4}{3}\binom{6}{1} + \binom{4}{4}\binom{6}{0} \\ &= 6 \cdot 15 + 4 \cdot 6 + 1 \cdot 1 \\ &= 90 + 24 + 1 \\ &= 115. \end{aligned}$$

The grand-prize event corresponds to exactly 4 matches, so

$$N(G) = \binom{4}{4}\binom{6}{0} = 1.$$

Since every grand-prize outcome is also a prize-winning outcome, we have $G \subseteq E$, and therefore

$$P(G \mid E) = \frac{N(G)}{N(E)} = \frac{1}{115}.$$

Therefore, $m = 1$ and $n = 115$, which gives

$$m + n = 116.$$

Final Answer
The conditional probability is $\frac{1}{115}$. Hence, $m = 1$, $n = 115$, and the final answer is $\boxed{116}$.

Although no external agent is explicitly invoked at inference time, the model reveals the corresponding abilities learned during training:

- **Verification-like behavior:** Check whether the grand-prize event is a subset of the prize event.
- **Knowledge-like behavior:** Recall the combinatorial counting rule for exactly k matches.
- **Reasoning-like behavior:** Compute the number of outcomes corresponding to $k = 2, 3, 4$ matches.

D DISCUSSIONS

D.1 MORE DISCUSSIONS ON RELATED WORK

The work most closely related to AIPO is recent research on external-guidance-enhanced reinforcement learning and distillation (Zhang et al., 2025c; Lv et al., 2025; Zhang et al., 2025b; Chen et al., 2025b; Sun et al., 2025b; Lin et al., 2025; Ma et al., 2025; Fu et al., 2025; Zhao et al., 2026). Existing methods mainly incorporate external information through supervised learning on expert trajectories, token-level supervision from teacher models, or RL objectives that learn from off-policy expert trajectories. These approaches differ from AIPO in when external guidance enters the learning process. AIPO introduces collaborator feedback directly during rollout, conditioned on the policy's current partial trajectory and reasoning bottleneck, thereby constructing mixed-policy trajectories that may be unlikely to arise from unaided on-policy exploration. Learning from these trajectories is supported by the corrected importance sampling coefficient and clipping strategy introduced in § 3.2.

Comparison with On-Policy Distillation. On-policy distillation (Zhao et al., 2026) preserves student-generated rollouts while using teacher logits as token-level supervision. It therefore improves how the student learns from trajectories that have already been generated, but does not directly modify the states visited during rollout. In contrast, AIPO introduces external feedback inside the rollout itself. The policy identifies an intermediate bottleneck and queries a collaborator conditioned on its current reasoning state, so the external guidance directly affects the resulting trajectory distribution. A more detailed discussion is provided in § D.4. We further discuss the distinction from SFT followed by RL in § D.6.

Comparison with Multi-Agent Guidance. Among existing methods, Lin et al. (2025) is particularly related because it also introduces multiple agents during rollout for communication and feedback. However, AIPO differs in two aspects. First, interaction is initiated adaptively by the policy at intermediate reasoning bottlenecks, allowing external assistance to be conditioned on the current partial trajectory. Second, AIPO explicitly optimizes mixed-policy trajectories containing both policy-generated and collaborator-generated tokens, whereas Lin et al. (2025) optimizes the policy primarily through on-policy trajectories. This distinction allows AIPO to use external collaborators as part of the exploration process rather than only as auxiliary sources of feedback.

Comparison with Self-Reflection and Self-Evolution. Self-reflection and self-evolution methods (Shinn et al., 2023; Dou et al., 2024) typically rely on feedback, critiques, or revisions generated by the policy model itself. Their exploration is therefore still primarily determined by information accessible to the model. In contrast, AIPO obtains additional reasoning signals from external collaborators during training, which can introduce knowledge or reasoning patterns that are unlikely under the current policy distribution. Importantly, policy optimization in AIPO remains grounded in verifiable task rewards rather than relying solely on self-generated supervision.

D.2 Mitigating Errors in Active Interactions

While AIPO achieves significant performance improvements, active interaction with collaborators may introduce erroneous information, due to hallucinations (Huang et al., 2025). Such errors can propagate through the reasoning process and potentially mislead the policy model. However, AIPO optimizes the policy model toward maximizing expected rewards defined by verifiable outcome correctness, providing a stable learning signal that consistently favors trajectories leading to correct solutions. Moreover, prior studies show that, when guided by reasonable and verifiable rewards, policy models can learn to filter, retrieve, and reflect on information, thereby extracting useful signals from noisy or imperfect inputs (Bensal et al., 2025; Kumar et al., 2025; Zhang et al., 2025a; Wei et al., 2025; DeepSeek-AI et al., 2025). Therefore, although active interaction introduces the risk of error propagation, it also expands the exploration space and provides richer learning opportunities. The RLVR objective enables the policy model to exploit these benefits while progressively mitigating the negative effects of unreliable collaborator feedback.

D.3 Why AIPO Does Not Require Distributionally Matched Collaborators

Equation (5) contains no term depending on π_ϵ, which may appear to imply that the objective is indifferent to which collaborator is used. We clarify what this property does and does not mean.

Importance sampling imposes no formal constraint on the relation between behaviour and target policy, but it is only well behaved in practice when the two are not too dissimilar: a large discrepancy produces extreme importance weights and destabilizes optimization. This is precisely the failure mode our approximation is designed to avoid. Rather than estimating the collaborator's token-level distribution—which is often inaccessible, tokenizer-incompatible, or expensive to evaluate—we approximate it by a one-hot distribution on the sampled external token, so the coefficient reduces to the probability the *policy* assigns to that token. This quantity is bounded in $[0, 1]$ by construction, and the pathological weights of conventional importance sampling under strong distribution mismatch cannot arise.

Two consequences follow. First, AIPO needs neither collaborator log-probabilities nor a shared vocabulary, so it can learn from a broad range of external agents—larger models, differently trained models, and agents specialized in verification, knowledge retrieval, or sub-problem solving. Second, the lower-bound clip of Equation (6) ensures that informative external tokens are not discarded merely

because the policy currently assigns them low probability, which is exactly the situation created by a large policy–collaborator gap.

This robustness should not be read as a guarantee that any collaborator helps. What AIPO relaxes is a *compatibility* requirement: the collaborator need not resemble the policy distribution. It does not relax the *quality* requirement: feedback that is systematically irrelevant, or adversarially misleading, still injects unhelpful tokens into the trajectory. Two mechanisms limit the damage—the verifiable outcome reward down-weights trajectories that fail regardless of their external content (§ D.2), and the advantage term $\tilde{A}_t$ can be negative for such tokens—but we do not claim robustness to an adversarial collaborator, and we state this explicitly in § E.

D.4 Comparison with On-Policy Distillation

On-policy distillation (OPD) keeps trajectory generation on-policy while supervising it with teacher token-level distributions, and can be combined with RL so the student also optimizes task-level rewards. Its limitation is structural rather than a matter of supervision strength: guidance and exploration remain decoupled. The teacher scores trajectories that the *unaided* policy has already produced, so the trajectory distribution is still determined by the policy's unaided rollout behaviour, and the supervision signal is static with respect to where the policy actually got stuck. When teacher and student behaviours diverge substantially, token-level matching can also produce ineffective or negative transfer.

AIPO intervenes inside the rollout instead. The policy detects an intermediate bottleneck, chooses a collaborator, and receives feedback conditioned on its current partial trajectory. The resulting mixed-policy trajectory is one the policy could not have generated alone. Three consequences distinguish it from OPD. First, AIPO introduces reasoning directions that are unlikely to be discovered by independent exploration, so it changes which trajectories exist rather than only how they are scored. Second, guidance is targeted and state-dependent, and is requested only where it is needed, which makes it more sample-efficient than supervising every token of every trajectory. Third, the policy must *integrate* the feedback into its own reasoning to obtain reward, rather than reproduce a teacher sequence, and it operates without collaborators at inference.

Empirically, Table 1 compares AIPO against OPSD, our OPD-style baseline, under identical data, optimizer, and budget: AIPO is ahead on all eight benchmarks for both policy models and both collaborator choices.

D.5 Can the Policy Model Surpass its Collaborators?

Whether the trained policy can exceed the collaborator depends on the source of the capability gap.

When the collaborator is a substantially larger and more extensively trained model—for example Qwen3-30B-A3B-Instruct-2507 assisting Qwen2.5-7B-Instruct (Table 4)—the policy generally does not overtake it within our training budget. The gap there stems from model scale, pretraining data, and post-training budget, none of which interaction can supply.

When the collaborator shares the policy's backbone, the trained policy *can* surpass the standalone collaborator. The comparison is not between two equal models: the collaborators are used in three narrow functional roles—verification, knowledge synthesis, sub-problem solving—whose outputs the policy must combine, while the policy additionally receives task-level verifiable rewards. It can therefore internalize a composition of capabilities that no single role provides, and the same-backbone blocks of Table 1 show exactly this: AIPO with Qwen2.5-7B collaborators reaches 26.5 on AIME24 against 9.8 for the untrained shared backbone, and also exceeds every RL baseline trained without interaction.

D.6 Comparison with SFT Initialization followed by RL

A natural alternative is to generate expert data by multi-agent collaboration once, perform SFT, and then run RL. This is a reasonable initialization strategy, but it does not address the bottleneck AIPO targets.

The purpose of AIPO is not to introduce expert demonstrations but to change the exploration process itself through proactive, state-dependent interaction. SFT supervision is static and fixed before RL begins, so it cannot respond to where the policy fails *during* training. It encourages imitation of complete expert behaviour without identifying which intermediate decision resolves the student's current failure, and large teacher–student discrepancies make such imitation less effective. Once RL starts, exploration is again bounded by the policy's own distribution.

Our SFT and OPSD baselines instantiate the two closest variants of this idea—SFT on rejection-sampled expert trajectories, and teacher supervision injected during on-policy training—and both are outperformed by AIPO under matched budgets (Table 1). We read this as evidence that the gain comes from active, adaptive, bottleneck-conditioned interaction rather than from access to expert content per se. The two are nevertheless complementary: SFT initialization could be combined with AIPO, and we leave this to future work.

E Limitations and Future Work

While AIPO achieves significant performance improvements, several areas warrant further exploration. First, our method is currently limited to verifiable questions with definitive answers. Training on open-ended questions remains an open problem in the community and is beyond the scope of this paper. Second, due to computational resource constraints, we conducted experiments only on LLMs up to 32B parameters. Given that the scaling law (Hestness et al., 2017; Henighan et al., 2020) is an important principle in the field of LLMs, investigating the performance of AIPO on larger-scale LLMs will be another promising direction. Thirdly, the current scope of our experimentation and analysis is limited exclusively to the English language. Therefore, the applicability and performance of AIPO with interfaces and information presented in languages other than English remain an open question. Addressing this linguistic limitation is crucial for establishing the generalizability of the proposed framework across diverse linguistic contexts. Fourth, AIPO relaxes the requirement that the collaborator be distributionally similar to the policy, but not the requirement that its feedback be useful. A collaborator whose responses are systematically irrelevant or adversarial can still inject unhelpful tokens into the trajectory. The verifiable reward limits, but does not eliminate, the resulting damage, and we have not characterized AIPO's behaviour under an adversarial collaborator. Fifth, our evidence for boundary expansion concerns the *effective exploration boundary*: the set of problems that the trained policy can solve independently under a fixed inference-time sampling budget. Our DeepMath experiments, including out-of-distribution evaluation, provide evidence that this effective boundary can be broadened through training-time interaction. We do not interpret these experiments as establishing an expansion of the theoretical capability ceiling of the pretrained model. A rigorous test of such a stronger claim would require substantially larger evaluations and carefully constructed out-of-distribution problems (Yue et al., 2025a), which we leave to future work.