THE PATH MATTERS: TRAJECTORY-AWARE SEMI-SUPERVISED REINFORCEMENT LEARNING FOR LLM REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning with verifiable rewards (RLVR) improves the reasoning capabilities of large language models (LLMs), but its reliance on reference answers limits training when annotations are scarce. Recent semi-supervised methods introduce a small labeled set to filter reliable unlabeled data for subsequent reinforcement learning, but they do not fully exploit the value of the labeled data. Since related questions tend to admit similar reasoning trajectories, the verified trajectories of labeled examples can serve as references for assessing the reasoning produced on unlabeled samples. Motivated by this consideration, we propose STEER, a trajectory-aware semi-supervised reinforcement learning framework that extends sample-level reliability assessment to trajectory-level policy optimization. STEER maintains a dynamic reasoning graph anchored by trajectories whose final answers have been verified as correct on labeled examples, and scores each unlabeled rollout by its similarity to the trajectories retrieved from related questions, yielding a trajectory-level reward that complements answer-based feedback. Meanwhile, reliable trajectories generated for unlabeled questions are added to the graph to further support subsequent training. Results on five mathematical reasoning benchmarks demonstrate the effectiveness of our method.

1 INTRODUCTION

Large language models (LLMs) have made substantial progress in reasoning, with reinforcement learning with verifiable rewards (RLVR) providing an effective way to improve their ability to solve complex problems (Shao et al., 2024; Guo et al., 2025). By checking generated answers against reference answers, RLVR supplies outcome feedback that guides the policy toward successful solutions. However, obtaining reliable reference answers for a large training corpus remains costly, limiting the use of this approach when annotations are scarce. Learning from abundant unlabeled

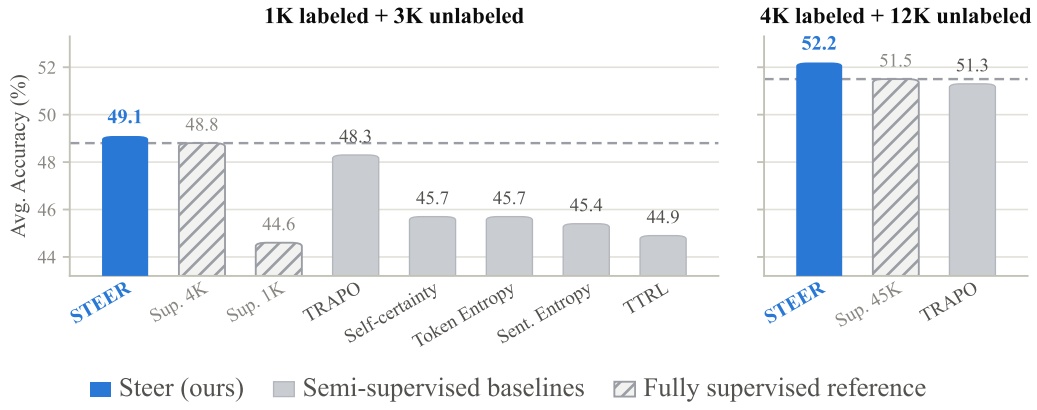

Figure 1: Results overview on the five mathematical reasoning benchmarks.

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.

questions with only a small amount of verified supervision is therefore an important direction for data-efficient reasoning optimization (Agarwal et al., 2025; Zuo et al., 2025; Yang et al., 2025).

To reduce reliance on labeled data, recent approaches construct rewards from the model's own predictions. Majority voting treats the most frequent sampled answer as a pseudo-label (Zuo et al., 2025), while entropy and self-certainty provide confidence-based objectives for policy optimization (Agarwal et al., 2025; Zhao et al., 2025). These approaches make unlabeled questions usable for reinforcement learning, but their feedback can reinforce incorrect predictions when confidence diverges from correctness. Semi-supervised RLVR introduces a small labeled set to anchor learning on the unlabeled pool. In particular, TRAPO records the evolution of prompt-level pass rates and selects unlabeled examples whose pass-rate curves resemble those of trusted examples, where the trusted set is progressively expanded with the examples accepted in earlier rounds (Yang et al., 2025).

However, existing semi-supervised RLVR methods exploit the scarce labeled examples only partially. Labeled examples are either appended to the training pool or used as a reference for prompt selection (Yang et al., 2025). Their verified reasoning trajectories, whose final answers are certified against reference answers, are left unused. Such trajectories constitute a reusable form of supervision, since related questions tend to admit similar reasoning trajectories. Although the answer to a mathematical question is instance-specific and cannot be transferred, related questions frequently rely on the same theorems, transformations, and solution structures. A rollout sampled for an unlabeled question is thus more likely to follow a viable reasoning path when it remains consistent with verified trajectories of related questions. Building on this observation, we organize the verified trajectories of trusted examples into a retrievable reference set and derive a trajectory-level score for each rollout of an unlabeled question, which modulates its group-relative advantage.

Motivated by this consideration, we propose STEER, i.e., **S**emi-supervised **T**rajectory-awar**E** r**E**inforcement lea**R**ning for LLM reasoning. Specifically, STEER maintains a dynamic reasoning graph that stores questions and accepted solution trajectories, initially anchored by labeled examples. For each unlabeled question, STEER samples multiple rollouts and retrieves related graph nodes based on question similarity. These rollouts provide two complementary signals for policy training. First, majority agreement among their answers determines the pseudo-label and a sample-level confidence weight. Second, each rollout is assigned a graph-based support score derived from its similarity to the retrieved trajectories of related questions. The confidence weight controls the unlabeled question's contribution to policy training, whereas the graph-based scores are normalized separately and used to augment the group-relative advantages of individual rollouts.

Our contributions are summarized as follows:

- ❶ **Limitation identification.** We identify that existing semi-supervised RLVR methods underexploit labeled examples by restricting their role to sample-level selection, leaving verified reasoning trajectories unused despite the tendency of related questions to share similar solution structures.
- ❷ **Proposed method.** We propose STEER, which maintains a dynamic reasoning graph that transfers support from accepted trajectories of related questions to individual unlabeled rollouts. The framework combines confidence-weighted GRPO with graph-based advantage shaping and independently controls pseudo-node admission to the graph.
- ❸ **Experimental evaluation.** We evaluate STEER on five mathematical reasoning benchmarks under multiple labeled–unlabeled data allocations of varying scale. Experimental results demonstrate that STEER consistently outperforms existing methods across all settings.

2 RELATED WORK

Semi-supervised learning. Semi-supervised learning leverages a small labeled set alongside a large unlabeled pool by exploiting structural assumptions about the data distribution (Chapelle et al., 2006; Liu et al., 2024; Xu et al., 2023). Temporal Ensembling (Laine & Aila, 2016) encourages prediction consistency across training epochs by maintaining exponential moving averages of model outputs as soft targets for unlabeled examples. MixMatch (Berthelot et al., 2019) unifies consistency regularization and entropy minimization by constructing mixed training examples from both labeled and unlabeled data, while FixMatch (Sohn et al., 2020) further simplifies this pipeline by retaining pseudo-labels only when the model's confidence on a weakly augmented view exceeds a fixed

threshold. These methods are highly effective in classification, where all examples share a fixed label space and inputs with similar features tend to share the same class.

Reinforcement learning under limited supervision. RLVR has become a central approach for improving mathematical reasoning in language models by sampling multiple responses and using rule-based verifiers to supply outcome rewards for group-relative policy optimization (Shao et al., 2024; Guo et al., 2025). Its primary bottleneck is the requirement for ground-truth answers for every training prompt. To relax this requirement, unsupervised variants replace external labels with signals derived from the model's own predictions. Agarwal et al. (2025) show that minimizing the entropy of the output distribution over unlabeled questions yields consistent reasoning improvements, while Zhao et al. (2025) propose self-certainty as a reward signal that does not rely on any reference answer. TTRL (Zuo et al., 2025) treats the majority answer among sampled responses as a pseudo-label and applies standard RLVR on top of it. However, these signals conflate model confidence with correctness, and an incorrect majority can be reinforced, leading to instability or collapse when optimized without an external anchor (Zhang et al., 2025). Semi-supervised RLVR addresses this by introducing a small labeled set. TraPO (Yang et al., 2025) records the evolution of prompt-level pass rates over training and selects unlabeled examples whose pass-rate curves resemble those of trusted labeled examples, with the trusted set progressively expanded in each round.

3 METHOD

In this section, we present STEER, a graph-aware semi-supervised reinforcement learning framework for LLM reasoning. STEER uses answer agreement among the sampled rollouts as a sample-level confidence that weights how much an unlabeled sample contributes to training. To supervise reasoning quality more finely, we further score the quality of each individual rollout against the verified reasoning trajectories stored in a dynamic graph of reliable samples, and finally combine the two complementary scores to train the model with GRPO (Shao et al., 2024; Guo et al., 2025). An overview of the overall framework is shown in Figure 2.

3.1 PROBLEM FORMULATION

Let Σ be the vocabulary of the language model, and let $\mathcal{X}$ and $\mathcal{Y}$ be the sets of prompts and responses, both consisting of finite token sequences over Σ. A policy π_θ maps a prompt $\boldsymbol{x} \in \mathcal{X}$ to a distribution over responses and generates $\boldsymbol{y} = (y^1, \ldots, y^{|\boldsymbol{y}|}) \in \mathcal{Y}$ autoregressively. Training uses a small labeled set together with a much larger set of prompts whose answers are withheld,

$$\mathcal{D}_{\mathrm{L}} = \big\{(\boldsymbol{x}_i, a_i^\star)\big\}_{i=1}^{N_{\mathrm{L}}}, \qquad \mathcal{D}_{\mathrm{U}} = \{\boldsymbol{x}_i\}_{i=1}^{N_{\mathrm{U}}}, \qquad N_{\mathrm{U}} \gg N_{\mathrm{L}}, \tag{1}$$

where $a_i^\star$ is the verified reference answer of the labeled prompt $\boldsymbol{x}_i$.

Every response consists of a reasoning trajectory followed by a final answer. Let $\mathcal{A}$ be the set of answer strings, and let $h : \mathcal{Y} \to \mathcal{A}$ be the operator that extracts the final answer from a response. Two answers are compared by the indicator $\mathrm{Eq} : \mathcal{A} \times \mathcal{A} \to \{0, 1\}$, where $\mathrm{Eq}(a, a') = 1$ if a and a' are mathematically equivalent under rule-based verification and $\mathrm{Eq}(a, a') = 0$ otherwise. Verification is the only source of ground-truth feedback and is available for labeled prompts only.

At optimization step t we sample a batch $\mathcal{B}_t$ of B prompts from $\mathcal{D}_{\mathrm{L}} \cup \mathcal{D}_{\mathrm{U}}$ and draw G responses per prompt from the current policy,

$$\mathcal{O}_i = \{\boldsymbol{y}_{i,1}, \ldots, \boldsymbol{y}_{i,K}\}, \qquad \boldsymbol{y}_{i,k} \sim \pi_\theta(\cdot \mid \boldsymbol{x}_i). \tag{2}$$

The training objective is to maximize the expected reward of the policy, often together with a Kullback–Leibler constraint that keeps the policy close to a reference model,

$$\max_\theta \; \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D},\, \boldsymbol{y}\sim\pi_\theta(\cdot|\boldsymbol{x})}\big[r(\boldsymbol{x}, \boldsymbol{y})\big] - \beta_{\mathrm{KL}}\, \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\Big[D_{\mathrm{KL}}\big(\pi_\theta(\cdot \mid \boldsymbol{x}) \,\|\, \pi_{\mathrm{ref}}(\cdot \mid \boldsymbol{x})\big)\Big], \tag{3}$$

where $r : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is the reward function that scores how desirable a response is for a given prompt, π_{ref} is a frozen reference policy, and $\beta_{\mathrm{KL}} \geq 0$ controls the strength of the constraint. Under RLVR, r requires no learned reward model and is instead instantiated by rule-based verification against the target answer of the prompt, so that a response is rewarded only when its extracted answer $h(\boldsymbol{y})$ is equivalent to that target. Since verification is available for labeled prompts only, this reward cannot be evaluated on $\mathcal{D}_{\mathrm{U}}$, which motivates the surrogate targets introduced below.

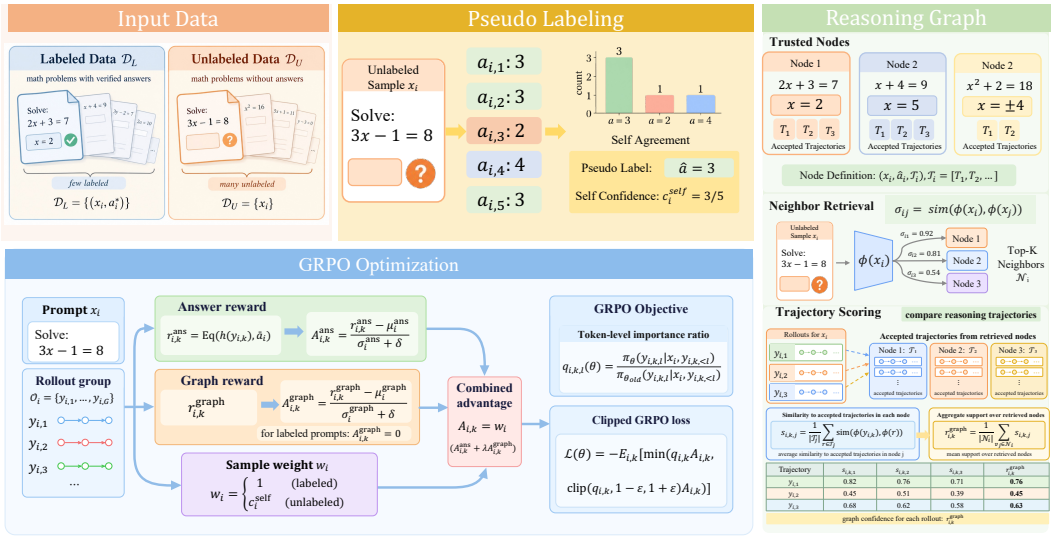

Figure 2: Overview of STEER, which combines answer agreement with trajectory-level scores retrieved from a dynamic reasoning graph to optimize the policy on unlabeled data.

3.2 PSEUDO-LABELS FROM ANSWER AGREEMENT

For an unlabeled prompt we partition its group into answer equivalence classes,

$$n_i(a) = \sum_{k=1}^{G} \mathrm{Eq}\big(h(\boldsymbol{y}_{i,k}), a\big), \qquad a \in \mathcal{A}. \tag{4}$$

The pseudo-label is the most supported answer, and its vote share defines the sample-level self-confidence of the prompt,

$$\hat{a}_i = \arg\max_a \, n_i(a), \qquad c_i^{\mathrm{self}} = \frac{1}{G} \max_a \, n_i(a) \in (0, 1]. \tag{5}$$

We use the group only when this maximizer is unique. Let $\mathcal{C}_t$ be the set of unlabeled prompts of batch $\mathcal{B}_t$ whose majority answer is unique. To simplify notation, we write the target answer of a prompt as

$$\bar{a}_i = \begin{cases} a_i^\star, & \boldsymbol{x}_i \in \mathcal{D}_{\mathrm{L}}, \\ \hat{a}_i, & \boldsymbol{x}_i \in \mathcal{D}_{\mathrm{U}}, \end{cases} \qquad \mathcal{T}_i = \big\{\boldsymbol{y}_{i,k} \in \mathcal{O}_i : \mathrm{Eq}(h(\boldsymbol{y}_{i,k}), \bar{a}_i) = 1\big\}, \tag{6}$$

where $\mathcal{T}_i$ collects the *accepted trajectories* of the prompt. For a labeled prompt these trajectories are verified against the reference answer; for an unlabeled prompt they are the members of the majority class and are only as reliable as the pseudo-label they agree with.

3.3 DYNAMIC REASONING GRAPH

The graph at step t is a set of nodes $\mathcal{V}_t$, where each node holds a prompt whose answer is currently trusted together with trajectories that reach that answer,

$$v_j = \big(\boldsymbol{x}_j,\ a_j,\ \mathcal{T}_j\big), \qquad \mathcal{T}_j \subseteq \mathcal{Y}, \tag{7}$$

where $\boldsymbol{x}_j$ is the prompt, a_j its reference or pseudo answer, and $\mathcal{T}_j$ the accepted trajectories that support that answer. A node is refreshed only when its prompt appears in the current batch, in which case $\mathcal{T}_j$ is recomputed from the new group by Eq. (6).

Similar questions share similar solution strategies, which are reflected in the reasoning trajectories produced by the model. For each unlabeled sample we therefore retrieve related questions by question similarity and measure the similarity between its rollouts and the trajectories of those questions, which yields a graph score that assesses the reliability of the reasoning of each rollout. Let ϕ be a text

Algorithm 1 Training procedure of STEER

Require: labeled set $\mathcal{D}_{\mathrm{L}}$, unlabeled set $\mathcal{D}_{\mathrm{U}}$, initial policy π_θ, steps T, warm-up T_{w}, group size G, neighborhood size K, thresholds $\tau_{\mathrm{s}} \le \tau$, coefficient λ
Ensure: optimized policy π_θ

$\mathcal{V}_1 \leftarrow \{(\boldsymbol{x}_j, a_j^\star, \emptyset) : \boldsymbol{x}_j \in \mathcal{D}_{\mathrm{L}}\}$ ▷ labeled prompts as initial nodes
for $t = 1$ **to** T **do**
  sample a batch $\mathcal{B}_t$ of B prompts from $\mathcal{D}_{\mathrm{L}}$ if $t \le T_{\mathrm{w}}$, otherwise from $\mathcal{D}_{\mathrm{L}} \cup \mathcal{D}_{\mathrm{U}}$
  for each prompt $\boldsymbol{x}_i \in \mathcal{B}_t$ **do**
    sample rollouts $\mathcal{O}_i = \{\boldsymbol{y}_{i,k}\}_{k=1}^{G} \sim \pi_\theta(\cdot \mid \boldsymbol{x}_i)$ ▷ Eq. (2)
    if $\boldsymbol{x}_i \in \mathcal{D}_{\mathrm{L}}$ **then**
      $\bar{a}_i \leftarrow a_i^\star,\ w_i \leftarrow 1,\ A_{i,k}^{\mathrm{graph}} \leftarrow 0$
    else
      discard $\boldsymbol{x}_i$ unless its majority answer is unique and $c_i^{\mathrm{self}} \ge \tau_{\mathrm{s}}$ ▷ Eq. (5)
      $\bar{a}_i \leftarrow \hat{a}_i,\ w_i \leftarrow c_i^{\mathrm{self}}$
      retrieve $\mathcal{N}_i$ from $\mathcal{V}_t$ and score every rollout by $r_{i,k}^{\mathrm{graph}}$ ▷ Eqs. (9), (11)
    end if
    $\mathcal{T}_i \leftarrow \{\boldsymbol{y}_{i,k} : \mathrm{Eq}(h(\boldsymbol{y}_{i,k}), \bar{a}_i) = 1\},\ r_{i,k}^{\mathrm{ans}} \leftarrow \mathrm{Eq}(h(\boldsymbol{y}_{i,k}), \bar{a}_i)$ ▷ Eqs. (6), (13)
  end for
  $A_{i,k} \leftarrow w_i\big(A_{i,k}^{\mathrm{ans}} + \lambda A_{i,k}^{\mathrm{graph}}\big)$ for all rollouts in the batch ▷ Eqs. (14)–(16)
  update θ by minimizing $\mathcal{L}(\theta)$ ▷ Eq. (18)
  refresh $\mathcal{T}_j$ of every node whose prompt appears in $\mathcal{B}_t$
  $\mathcal{V}_{t+1} \leftarrow \mathcal{V}_t \cup \big\{(\boldsymbol{x}_i, \hat{a}_i, \mathcal{T}_i) : \boldsymbol{x}_i \in \mathcal{B}_t \cap \mathcal{D}_{\mathrm{U}},\ c_i^{\mathrm{self}} \ge \tau\big\}$ ▷ Eq. (12)
end for

encoder that maps a question or a reasoning trajectory into a vector space. The similarity between a prompt $\boldsymbol{x}_i$ and the question of a node v_j is defined as:

$$\sigma_{ij} = \mathrm{sim}\big(\phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j)\big), \qquad \mathrm{sim}(\boldsymbol{u}, \boldsymbol{v}) = \max\big\{0, \cos(\boldsymbol{u}, \boldsymbol{v})\big\}, \tag{8}$$

where $\cos(\cdot, \cdot)$ is the cosine similarity. The neighborhood retrieved for $\boldsymbol{x}_i$ is then

$$\mathcal{N}_i = \mathrm{Top}_K\Big(\big\{v_j \in \mathcal{V}_t :\ j \ne i\big\};\ \sigma_{ij}\Big), \tag{9}$$

where $\mathrm{Top}_K(\mathcal{S}; \sigma)$ denotes the at most K elements of $\mathcal{S}$ with the largest score.

Each node in $\mathcal{N}_i$ stores trajectories that already reach a trusted answer, and we use these trajectories as references against which the rollouts of $\boldsymbol{x}_i$ are scored. The agreement between rollout $\boldsymbol{y}_{i,k}$ and a node v_j is its mean similarity to the trajectories stored in that node,

$$s_{i,k,j} = \frac{1}{|\mathcal{T}_j|} \sum_{\boldsymbol{r} \in \mathcal{T}_j} \mathrm{sim}\big(\phi(\boldsymbol{y}_{i,k}), \phi(\boldsymbol{r})\big), \tag{10}$$

and the graph score of the rollout is the mean agreement over its retrieved nodes,

$$r_{i,k}^{\mathrm{graph}} = \frac{1}{|\mathcal{N}_i|} \sum_{v_j \in \mathcal{N}_i} s_{i,k,j}. \tag{11}$$

When the self-confidence of an unlabeled sample is sufficiently high, the sample is regarded as reliable as well and is inserted into the graph as a pseudo-labeled node. A candidate whose self-confidence reaches a threshold τ is inserted as a new node carrying its pseudo-label and its accepted trajectories,

$$\mathcal{V}_{t+1} = \mathcal{V}_t \cup \big\{(\boldsymbol{x}_i, \hat{a}_i, \mathcal{T}_i) :\ \boldsymbol{x}_i \in \mathcal{C}_t,\ c_i^{\mathrm{self}} \ge \tau\big\}. \tag{12}$$

The graph is initialized with all labeled samples and the trajectories collected during a warm-up of T_{w} steps, throughout which the policy is trained on labeled data only.

3.4 CONFIDENCE-DECOUPLED GROUP-RELATIVE OPTIMIZATION

The outcome reward of a rollout is its agreement with the target answer defined in Eq. (6),

$$r_{i,k}^{\mathrm{ans}} = \mathrm{Eq}\big(h(\boldsymbol{y}_{i,k}), \bar{a}_i\big) \in \{0, 1\}. \tag{13}$$

Following GRPO, rewards are standardized within each group,

$$A_{i,k}^{\text{ans}} = \frac{r_{i,k}^{\text{ans}} - \mu_i^{\text{ans}}}{\sigma_i^{\text{ans}} + \delta}, \qquad \mu_i^{\text{ans}} = \frac{1}{G}\sum_{k=1}^{G} r_{i,k}^{\text{ans}}, \qquad \sigma_i^{\text{ans}} = \sqrt{\frac{1}{G}\sum_{k=1}^{G}\big(r_{i,k}^{\text{ans}} - \mu_i^{\text{ans}}\big)^2}, \tag{14}$$

where $\delta > 0$ ensures numerical stability. The resulting advantage is identical for all rollouts that reach the target answer and therefore provides no means of distinguishing their reasoning.

In addition to the outcome reward, we add a graph reward that assesses the reliability of each trajectory. For unlabeled prompts we accordingly standardize the graph scores within the group,

$$A_{i,k}^{\text{graph}} = \frac{r_{i,k}^{\text{graph}} - \mu_i^{\text{graph}}}{\sigma_i^{\text{graph}} + \delta}, \tag{15}$$

where μ_i^{graph} and σ_i^{graph} denote the mean and standard deviation of $\{r_{i,k}^{\text{graph}}\}_{k=1}^{G}$, and we set $A_{i,k}^{\text{graph}} = 0$ for labeled prompts, whose feedback is already verified.

The two standardized signals are combined as

$$A_{i,k} = w_i\Big(A_{i,k}^{\text{ans}} + \lambda A_{i,k}^{\text{graph}}\Big), \qquad w_i = \begin{cases} 1, & \boldsymbol{x}_i \in \mathcal{D}_{\text{L}}, \\ c_i^{\text{self}}, & \boldsymbol{x}_i \in \mathcal{D}_{\text{U}}, \end{cases} \tag{16}$$

where $\lambda \geq 0$ is a hyperparameter that controls the strength of the graph reward.

Let $\pi_{\theta_{\text{ref}}}$ denote the behavior policy that produced the rollouts and let

$$q_{i,k,l}(\theta) = \frac{\pi_\theta\big(y_{i,k}^l \mid \boldsymbol{x}_i, y_{i,k}^{<l}\big)}{\pi_{\theta_{\text{ref}}}\big(y_{i,k}^l \mid \boldsymbol{x}_i, y_{i,k}^{<l}\big)} \tag{17}$$

be the importance ratio at token position l. The policy is optimized by minimizing the clipped surrogate loss with a reference-policy regularizer,

$$\begin{aligned}\mathcal{L}(\theta) = &- \mathbb{E}_{i,k,l}\Big[\min\big(q_{i,k,l}(\theta)A_{i,k},\ \text{clip}\big(q_{i,k,l}(\theta), 1-\varepsilon, 1+\varepsilon\big)A_{i,k}\big)\Big] \\ &+ \beta_{\text{KL}}\,\mathbb{E}_{i,k,l}\Big[D_{\text{KL}}\big(\pi_\theta(\cdot \mid \boldsymbol{x}_i, y_{i,k}^{<l}) \,\|\, \pi_{\text{ref}}(\cdot \mid \boldsymbol{x}_i, y_{i,k}^{<l})\big)\Big],\end{aligned} \tag{18}$$

where $\varepsilon > 0$ is the clipping range, so that the loss instantiates the general objective in Eq. (3). Algorithm 1 summarizes the training procedure of STEER, covering rollout sampling, pseudo-labeling, graph retrieval and scoring, advantage computation, and graph expansion.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Training data. Following prior work (Yan et al., 2025; Yang et al., 2025), we train on OpenR1-Math-220k (Hugging Face, 2025).[1] The dataset contains roughly 220K competition-style mathematics problems drawn from NuminaMath 1.5 (LI et al., 2024), each paired with two to four reasoning traces distilled from DeepSeek-R1 (Guo et al., 2025). Traces are filtered for correctness against the reference answers, primarily with the rule-based Math Verify checker and, for a small fraction of problems that it cannot parse, with an LLM judge, so that every retained problem is accompanied by at least one verified solution. We construct three labeled–unlabeled allocations of increasing scale by sampling disjoint subsets from this pool: 1K labeled with 1K unlabeled, 1K labeled with 3K unlabeled, and 4K labeled with 12K unlabeled. Unlabeled data are obtained by masking the reference answers of the sampled problems, so that only the questions remain visible to the learner.

[1] https://github.com/huggingface/open-r1

Table 1: Performance on five mathematical reasoning benchmarks based on Qwen2.5-Math-7B under three different training paradigms. Rows in grey are fully supervised references.

Model	AIME	AMC	MATH-500	Minerva	Olympiad	Avg.
Original Models						
Qwen-Base	11.5	31.3	43.6	7.4	15.6	21.9
Qwen-Instruct	12.5	48.5	80.4	32.7	41.0	43.0
Unsupervised Methods Trained on 45K Samples w/o Any Labels						
TTRL	14.1	51.5	76.6	33.8	40.3	43.3
Self-certainty	16.9	51.7	77.6	34.9	38.8	44.0
Token-level Entropy	15.0	50.3	75.2	36.8	38.4	43.1
Sentence-level Entropy	11.4	42.1	68.0	32.7	30.5	36.9
Semi-supervised Methods Trained on 1K Labeled Samples & 3K Unlabeled Samples						
Fully Supervised w/ *1K* Labels	14.2	52.6	80.2	34.9	40.9	44.6
TTRL	14.9	55.3	77.8	33.1	43.6	44.9
Self-certainty	16.5	55.6	79.8	35.3	41.2	45.7
Token-level Entropy	18.2	53.4	80.2	34.6	41.9	45.7
Sentence-level Entropy	15.4	54.9	79.4	36.0	41.2	45.4
TRAPO	17.9	58.7	81.4	38.2	45.5	48.3
Fully Supervised w/ *4K* Labels	19.6	57.9	80.6	39.3	46.5	48.8
STEER (ours)	18.6	60.1	82.6	38.2	45.9	49.1
Semi-supervised Methods Trained on 4K Labeled Samples & 12K Unlabeled Samples						
TRAPO	24.3	60.0	84.6	39.3	48.3	51.3
Fully Supervised w/ *45K* Labels	25.1	62.0	84.4	39.3	46.8	51.5
STEER (ours)	24.8	61.8	85.4	40.1	49.0	52.2

Evaluation benchmarks and metrics. We evaluate on five mathematical reasoning benchmarks that span a broad range of difficulty: AIME, AMC (LI et al., 2024), Minerva (Lewkowycz et al., 2022), OlympiadBench (He et al., 2024), and MATH-500 (Hendrycks et al., 2021; Lightman et al., 2024). AIME consists of the 30 problems from the 2024 American Invitational Mathematics Examination, and AMC collects 83 problems from the 2022 and 2023 American Mathematics Competitions, all requiring an integer answer, together with the olympiad-level subset of OlympiadBench. Minerva comprises undergraduate-level quantitative reasoning problems in science and engineering, while MATH-500 is the widely used 500-problem subset of the MATH test set (Hendrycks et al., 2021) introduced by Lightman et al. (2024) and covers competition mathematics across five difficulty levels. We report the unweighted mean of the five benchmark scores in the Avg. column.

Baselines. We compare STEER against supervised, unsupervised, and semi-supervised RLVR methods. For supervised training, we apply GRPO (Shao et al., 2024) to the labeled subset alone, which isolates the contribution of the unlabeled pool. In the unsupervised setting, we discard the reference answers of both subsets and evaluate four approaches that derive their training signal entirely from the policy itself: (1) **TTRL** (Zuo et al., 2025), which takes the majority-voted answer of a rollout group as a pseudo-label; (2) **Self-Certainty** (Zhao et al., 2025), which rewards the average KL divergence from the uniform distribution over the vocabulary to the model's next-token distribution, thereby encouraging confident predictions; (3) **Token-Level Entropy** (Agarwal et al., 2025), which minimizes the per-token predictive entropy to promote consistency; and (4) **Sequence-Level Entropy** (Agarwal et al., 2025), which uses the negative sequence-level entropy as a reward and thereby maximizes the likelihood of the sampled response. For semi-supervised training, we form hybrid baselines by applying GRPO to the labeled subset and each of the above unsupervised objectives to the unlabeled subset. We additionally compare with TRAPO (Yang et al., 2025), the state-of-the-art semi-supervised RLVR method, which tracks prompt-level pass-rate trajectories and retains unlabeled prompts whose learning dynamics resemble those of verified examples.

Model and optimization. Unless stated otherwise, all experiments start from Qwen2.5-Math-7B (Yang et al., 2024), a base model that is widely adopted in the RLVR literature. All methods, including every baseline, are optimized with GRPO under an identical budget: $G = 8$ rollouts per prompt sampled at temperature 1.0, a rollout batch of 128 prompts, a mini-batch of 64 for each policy

Table 2: Performance on Qwen3-4B-Base with 1K labeled and either 1K or 3K unlabeled samples. The **Avg.** column is the mean over all five benchmarks.

Model	AIME	AMC	MATH-500	Minerva	Olympiad	Avg.
Original Model						
Qwen3-4B-Base	7.5	26.7	45.8	16.2	28.1	24.9
Semi-supervised Methods Trained on 1K Labeled Samples & 1K Unlabeled Samples						
TTRL	9.8	32.5	52.2	19.1	32.1	29.1
Self-certainty	14.1	39.1	54.6	21.0	38.8	33.5
Token-level Entropy	22.8	44.8	58.8	22.8	46.2	39.1
Sentence-level Entropy	18.5	44.2	58.0	22.8	43.4	37.4
TRAPO	16.7	56.6	80.6	39.3	44.1	47.5
STEER (ours)	36.7	55.4	85.0	39.7	49.0	**53.2**
Semi-supervised Methods Trained on 1K Labeled Samples & 3K Unlabeled Samples						
TTRL	11.9	36.9	54.2	24.6	37.6	33.0
Self-certainty	1.0	23.1	48.2	28.3	16.3	23.4
Token-level Entropy	16.6	42.2	57.6	24.3	41.0	36.3
Sentence-level Entropy	23.1	45.5	59.2	24.3	44.0	39.2
TRAPO	26.7	59.0	85.0	42.6	48.1	52.3
STEER (ours)	30.0	59.0	84.0	42.6	48.3	**52.8**

update, and a constant learning rate of 10^{-6} with ADAMW. Prompts and responses are truncated at 4096 tokens each, and rewards are produced by rule-based verification with Math-Verify without any format or length bonus. The policy is regularized towards the initial model by a low-variance KL loss with coefficient 10^{-3}, and no entropy bonus is added. We report avg@32 on AIME and AMC to reduce the variance induced by their small test sets, and pass@1 on the remaining benchmarks. For the components specific to STEER, questions and reasoning trajectories are encoded as ℓ_2-normalized TF-IDF vectors, and each unlabeled prompt retrieves at most $K = 8$ neighbors from the reasoning graph. The shared configuration and method-specific hyperparameters are provided in Appendix A, including Table 3. Training is implemented on top of veRL (Sheng et al., 2024) with vLLM (Kwon et al., 2023) as the rollout engine, and every run uses a single node of 8 GPUs.

4.2 MAIN RESULTS

Table 1 and Figure 1 show that STEER achieves the best five-benchmark average in both semi-supervised allocations. With 1K labeled and 3K unlabeled samples, STEER reaches 49.1 on average, exceeding Self-certainty and Token-level Entropy (both 45.7) by 3.4 points and TTRL by 4.2 points. The gains over Self-certainty include $+4.5$ on AMC and $+4.7$ on OlympiadBench. Compared with TRAPO, STEER improves the average by 0.8 points and wins on four of five benchmarks, including AMC ($+1.4$) and MATH-500 ($+1.2$), while matching it on Minerva. These results support the benefit of trajectory-level guidance beyond prompt-level filtering.

Among methods trained on 45K unlabeled questions, Self-certainty performs best at 44.0, only 1.0 point above Qwen-Instruct. Using under 9% as many questions and 1K labels, STEER exceeds it by 5.1 points, including $+8.4$ on AMC and $+7.1$ on OlympiadBench. Adding labels also improves Sentence-level Entropy from 36.9 to 45.4, but it remains 3.7 points below STEER, suggesting that labeled anchors help while leaving room for trajectory-level guidance.

With 1K labels, STEER improves over fully supervised training on the same labeled set by 4.5 points ($44.6 \rightarrow 49.1$) and matches the average of supervision on $4\times$ more labels (48.8). With 4K labeled and 12K unlabeled samples, it reaches 52.2, exceeding the 45K-label supervised reference (51.5) by 0.7 points with roughly $11\times$ fewer labels. It also outperforms TRAPO by 0.9 points on average and 1.8 on AMC at this scale, indicating that the gains persist as the unlabeled pool grows.

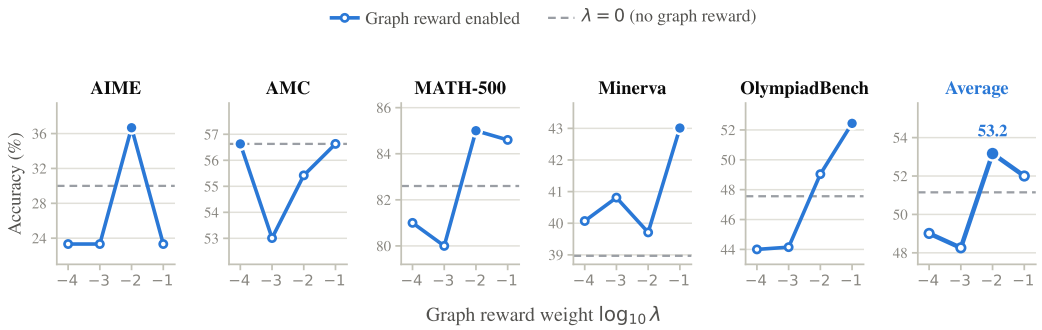

Figure 3: Sensitivity to the graph reward weight λ of Eq. (16) on Qwen3-4B-Base with 1K labeled and 1K unlabeled samples, per benchmark and on their average.

4.3 Further Analysis

Generalization to other backbones. We evaluate STEER on Qwen3-4B-Base using the same hyperparameters as for Qwen2.5-Math-7B. With 1K labeled and 3K unlabeled samples, Table 2 shows that STEER achieves the highest average score of 52.8, exceeding TRAPO by 0.5 points and the strongest confidence-based baseline, Sentence-level Entropy, by 13.6 points. Compared with TRAPO, it improves AIME by 3.3 points and OlympiadBench by 0.2, ties on AMC and Minerva, and trails by 1.0 point on MATH-500. The overall advantage on this backbone supports the transferability of trajectory-level guidance without backbone-specific hyperparameter tuning.

Effect of the labeled–unlabeled ratio. Keeping the labeled budget fixed at 1K, we also evaluate a 1:1 labeled–unlabeled ratio on Qwen3-4B-Base (Table 2). STEER achieves the best average at both ratios: 53.2 with 1K unlabeled samples and 52.8 with 3K, exceeding TRAPO by 5.7 and 0.5 points, respectively. The smaller unlabeled pool yields higher AIME accuracy (36.7 vs. 30.0), while the larger pool improves AMC and Minerva. Thus, STEER remains competitive across both allocations, although increasing the unlabeled pool does not improve every benchmark or the overall average.

Sensitivity to the graph reward weight λ. In Eq. (16) the coefficient λ sets how much of each update the graph advantage is allowed to drive. We sweep it over four orders of magnitude on Qwen3-4B-Base with 1K labeled and 1K unlabeled samples; Figure 3 reports each benchmark separately together with their average. The average response is not monotone. Accuracy peaks at $\lambda = 10^{-2}$ with 53.2, exceeding by 2.0 points the 51.1 obtained at $\lambda = 0$. A weight that is too small is not merely neutral but harmful: at $\lambda = 10^{-3}$ and 10^{-4} the run scores 48.3 and 49.0, that is 2.9 and 2.1 points *below* the no-graph reference. A graph term too weak to reorder the rollouts within a group perturbs the advantages without supplying usable discrimination, so the useful setting is bounded away from zero rather than approached in the limit. Only $\lambda \in [10^{-2}, 10^{-1}]$ improves on the reference (53.2 and 52.0), and that interval spans an order of magnitude, so the coefficient admits a wide operating range once it is inside it. We use $\lambda = 10^{-2}$ in all other experiments.

5 Conclusion

We presented STEER, a semi-supervised reinforcement learning framework that transfers guidance from limited labeled examples to individual unlabeled reasoning trajectories. By combining confidence-weighted policy optimization with support from a dynamic reasoning graph, STEER provides a training signal beyond answer agreement and sample-level reliability. Experiments on five mathematical reasoning benchmarks show consistent average improvements over the evaluated baselines across two backbones and multiple labeled–unlabeled allocations. On Qwen2.5-Math-7B, STEER matches the average performance of 4K-label supervision using only 1K labels and surpasses the 45K-label supervised reference using 4K labels. These results suggest that reusing trajectory-level support across related questions is a promising way to improve reasoning with limited annotation.

AI USE STATEMENT

We used large language models to polish the wording of this paper, including grammar and phrasing. The research topic, motivation, and technical content were conceived and written by the authors. Large language models also assisted in implementing parts of the method's code, and all generated code was reviewed and validated by the authors.

REFERENCES

Shivam Agarwal, Zimin Zhang, Lifan Yuan, Jiawei Han, and Hao Peng. The unreasonable effectiveness of entropy minimization in llm reasoning. *arXiv preprint arXiv:2505.15134*, 2025.

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A. Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, 2019.

Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien. *Semi-Supervised Learning*. Adaptive Computation and Machine Learning. MIT Press, 2006.

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

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024.

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 *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2021.

Hugging Face. Open R1: A fully open reproduction of DeepSeek-R1. `https://github.com/huggingface/open-r1`, 2025.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, 2023.

Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems*, 2022.

Jia LI, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. Numinamath. `[https://huggingface.co/AI-MO/NuminaMath-CoT](https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf)`, 2024.

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 *Proceedings of the International Conference on Learning Representations*, 2024.

Biao Liu, Ning Xu, Xiangyu Fang, and Xin Geng. Correlation-induced label prior for semi-supervised multi-label learning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 32224–32238, 2024.

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

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. *arXiv preprint arXiv:2409.19256*, 2024.

Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin Dogus Cubuk, Alexey Kurakin, Han Zhang, and Colin A. Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems*, 2020.

Ning Xu, Jun Shu, Renyi Zheng, Xin Geng, Deyu Meng, and Min-Ling Zhang. Variational label enhancement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):6537–6551, 2023.

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. Learning to reason under off-policy guidance. In *Advances in Neural Information Processing Systems*, 2025.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2.5-Math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

Shenzhi Yang, Guangcheng Zhu, Xing Zheng, Yingfan Ma, Zhongqi Chen, Bowen Song, Weiqiang Wang, Junbo Zhao, Gang Chen, and Haobo Wang. TraPO: A semi-supervised reinforcement learning framework for boosting llm reasoning. 2025.

Yanzhi Zhang, Zhaoxi Zhang, Haoxiang Guan, Yilin Cheng, Yitong Duan, Chen Wang, Yue Wang, Shuxin Zheng, and Jiyan He. No free lunch: Rethinking internal feedback for llm reasoning. *arXiv preprint arXiv:2506.17219*, 2025.

Xuandong Zhao, Zhewei Kang, Aosong Feng, Sergey Levine, and Dawn Song. Learning to reason without external rewards. *arXiv preprint arXiv:2505.19590*, 2025.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen Zhang, Xinwei Long, Ermo Hua, Biqing Qi, Youbang Sun, Zhiyuan Ma, Lifan Yuan, Ning Ding, and Bowen Zhou. TTRL: Test-time reinforcement learning. *arXiv preprint arXiv:2504.16084*, 2025.

Table 3: Training and evaluation configuration of STEER. The upper block is shared by all methods, including every baseline; the lower block collects the hyperparameters specific to STEER.

Group	Hyperparameter	Value
Model	Backbone	Qwen2.5-Math-7B
Data	Rollout batch size B	128
	Max prompt length	4096
	Max response length	4096
Optimization	Advantage estimator	GRPO
	Rollouts per prompt G	8
	Rollout temperature	1.0
	Mini-batch size	64
	Optimizer	ADAMW ($\beta = (0.9, 0.999)$)
	Learning rate	1×10^{-6} (constant)
	KL coefficient β_{KL}	1×10^{-3} (low-variance)
STEER	Warm-up epochs T_{w}	0
	Neighborhood size K	8
	Graph reward coefficient λ	1×10^{-2}

A TRAINING CONFIGURATION AND BASELINE HYPERPARAMETERS

All baselines share the backbone, optimizer, rollout budget, and evaluation protocol reported in Table 3, so that differences in performance are attributable to the training signal rather than to the compute budget. Table 3 also lists the hyperparameters specific to STEER; the settings below describe the individual baselines.

Unsupervised and hybrid baselines. TTRL, Self-Certainty, Token-Level Entropy, and Sequence-Level Entropy introduce no additional hyperparameters beyond the shared configuration: each replaces the verifiable reward on unlabeled prompts with its own objective and is otherwise optimized identically. For TTRL the pseudo-label is the majority answer of the same group of $G = 8$ rollouts that is used for the policy update, so no separate voting budget is required. In the hybrid semi-supervised variants, labeled and unlabeled prompts are drawn from the same batch, with the verifiable reward applied to the former and the unsupervised objective to the latter, and their losses are summed without additional weighting.

TRAPO. We reproduce TRAPO with the configuration reported by Yang et al. (2025). Its selection mechanism keeps the top-p fraction of unlabeled prompts at $p = 0.1$ and accepts a prompt when its trajectory-matching score exceeds $\Gamma = 0.4$. The warm-up phase, during which training uses labeled data only, lasts 8 epochs in the two in-distribution allocations (1K labeled with 3K unlabeled, and 4K labeled with 12K unlabeled) and 10 epochs in the setting that pairs 1K labeled in-distribution samples with 1K unlabeled out-of-distribution samples. Following the same work, the KL regularizer is disabled for this baseline ($\beta_{\mathrm{KL}} = 0$) and the entropy coefficient is set to 0.01.