# BOOSTING LLM EXPLORATION VIA WEAK-MODEL GUIDANCE IN RLVR

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Reinforcement Learning with Verifiable Rewards (RLVR) can improve LLM reasoning while reducing policy entropy and pass@$k$ performance at large $k$. We study whether partial reasoning trajectories from smaller, weaker models can counter this loss of coverage during RLVR training. The target model learns to continue from these auxiliary prefixes on a subset of training prompts while retaining question-only training on the others. Across two 7B target models and six mathematical reasoning benchmarks, this approach improves average large-$k$ pass rates over vanilla GRPO, although pass@1 and individual benchmark results are mixed. Training dynamics and prefix-quality analyses are consistent with a trajectory-perturbation explanation: unfamiliar prefixes are associated with higher early-stage policy entropy even when they provide little useful semantic guidance. These observations motivate weak-model prefixes as a lightweight exploration intervention without additional supervised fine-tuning or reward redesign.

## 1 INTRODUCTION

Reinforcement Learning with Verifiable Rewards (RLVR) has been widely adopted in the post-training stage for eliciting strong reasoning from Large Language Models (LLMs). Given an input prompt, the model generates a response and receives feedback from an external verifier. The training objective increases the probability of generating high-reward responses while penalizing low-reward ones, thereby enabling sequence-level optimization without requiring additional human-curated annotations (Schulman et al., 2017; Shao et al., 2024). RLVR has been extensively used in both open-source and closed-source LLMs to improve their ability to solve challenging reasoning tasks, as demonstrated by models such as DeepSeek-R1 (Guo et al., 2025), GPT-5 (Singh et al., 2025), and Gemini (Comanici et al., 2025). These models have achieved high accuracy on various mathematical reasoning benchmarks. However, most existing evaluations focus primarily on pass@1 performance, namely the accuracy achieved with a single trial, while largely overlooking reasoning coverage.

During RLVR training, the model is optimized to maximize the probability of generating reasoning trajectories that lead to correct answers. Cui et al. (2025) point out that LLMs tend to trade entropy for performance during reinforcement learning. Specifically, they observe that policy entropy decreases sharply in the early stage of training, which limits the model's ability to conduct broad exploration. As a result, once the model becomes over-confident after a few training steps, it may have fewer opportunities to discover alternative reasoning paths that could also lead to correct answers and subsequently increase their probabilities. Along similar lines, Yue et al. (2025) find that RLVR primarily increases the model's confidence in reasoning paths that have already been established, while narrowing the overall reasoning coverage. They support this conclusion using pass@$k$ metrics (Brown et al., 2024), where a problem is considered solved if the model produces a correct answer within $k$ sampled responses. When $k$ is sufficiently large, the base model can outperform the RLVR-trained model, suggesting that RLVR may improve single-sample accuracy at the cost of reduced solution diversity.

Existing approaches address the degradation of pass@$k$ through entropy regularization (Cui et al., 2025), gradient or reward design (Peng et al., 2025; Chen et al., 2025), and input-level changes such as prompt paraphrasing (Liang et al., 2025). Our work also intervenes at the input level, but uses incomplete trajectories from weaker models during RLVR training. This setting lets us ask whether



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.

guidance that is often semantically unreliable can nevertheless help recover some of the large-$k$ coverage lost under GRPO.

We investigate how the target model responds to partial trajectories generated by different model families. Such prefixes may present reasoning states that the target model would rarely generate on its own. Unlike conventional knowledge distillation, which typically transfers high-quality outputs from a stronger teacher (Park et al., 2019; Taori et al., 2023), our auxiliary models are smaller and their prefixes need not be correct. We hypothesize that continuing from these unfamiliar states can expose the target policy to alternative trajectories during training. This is a proposed explanation for the observed results, not a claim that prefix quality and distributional mismatch have been causally separated.

We evaluate our method against a standard RLVR baseline in which the model generates responses solely from the original question, without any reasoning-path prefix. Across the tested target models and mathematical reasoning benchmarks, the largest average gains occur at higher values of $k$; pass@1 and per-benchmark effects are less consistent. This pattern suggests that prefix guidance can partially recover the coverage lost under standard GRPO, even though the target model receives no auxiliary prefix at evaluation time.

Our main contributions are as follows:

- We examine how partial trajectories from different model families affect RLVR training, connecting auxiliary-prefix quality, target-policy entropy, and performance under repeated sampling. This analysis characterizes the behavior of a simple input-level intervention rather than introducing a new RL objective.
- We study a training-time intervention in which weak auxiliary models supply partial reasoning trajectories to a stronger target policy. The target model learns to continue these prefixes alongside question-only GRPO training, but requires no auxiliary model at evaluation time. Its clearest gains over vanilla GRPO appear in average pass@$k$ at larger $k$, while effects on pass@1 and individual benchmarks are mixed.
- We analyze training dynamics and the quality of prefixes from different auxiliary models. Many weak-model prefixes provide little or misleading semantic guidance, whereas higher-quality homologous prefixes do not yield comparable large-$k$ gains in the tested ablation. This contrast motivates a trajectory-level perturbation explanation, without establishing it as the sole causal mechanism.

## 2 Preliminaries on GRPO and Entropy

**Group Relative Policy Optimization (GRPO).** GRPO (Shao et al., 2024) is a reinforcement learning algorithm adapted from Proximal Policy Optimization (PPO) (Schulman et al., 2017). It is widely used in RLVR frameworks because it avoids training a separate value model and instead estimates advantages by comparing multiple sampled responses to the same prompt.

Let $\pi_\theta$ denote the policy model parameterized by $\theta$, and let $\pi_{\theta_{\text{old}}}$ denote the policy used to generate rollouts before the current update. Given a prompt $q$, GRPO samples a group of $G$ responses

$$\{r_i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(\cdot \mid q).$$

Each response $r_i$ is evaluated by a verifier or reward model, producing a scalar reward $R_i$. GRPO then computes a group-normalized advantage for each sampled output:

$$\hat{A}_i = \frac{R_i - \text{mean}(\{R_j\}_{j=1}^{G})}{\text{std}(\{R_j\}_{j=1}^{G}) + \varepsilon_{\text{adv}}}. \tag{1}$$

Here $\varepsilon_{\text{adv}} > 0$ prevents division by zero when all responses in a group receive the same reward. The same sequence-level advantage $\hat{A}_i$ is assigned to all tokens in the corresponding response $r_i$.

The GRPO objective adopts a PPO-style clipped surrogate loss with a KL penalty against a frozen reference model $\pi_{\text{ref}}$. Specifically, the objective is defined as

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{q,\{r_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}}\Bigg[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|r_i|}\sum_{k=1}^{|r_i|}\Bigg(\min\Big(\rho_{i,k}(\theta)\hat{A}_i, \text{clip}\left(\rho_{i,k}(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_i\Big) - \beta\,\mathbb{D}_{\text{KL}}\big(\pi_\theta \parallel \pi_{\text{ref}}\big)\Bigg)\Bigg],$$

where $\rho_{i,k}(\theta) = \frac{\pi_\theta(r_{i,k}|q,r_{i,<k})}{\pi_{\theta_{\text{old}}}(r_{i,k}|q,r_{i,<k})}$ is the token-level importance sampling ratio, $\epsilon$ is the clipping coefficient, and $\beta$ controls the strength of the KL regularization. In our experiments, $\beta = 0$, so the KL term is not used.

**Entropy Collapse in RLVR.** The training dynamics of GRPO reveal an inherent trade-off between exploitation and exploration. As the model increases the probability of generating high-reward responses, it also assigns greater confidence to its own preferred reasoning trajectories, often at the cost of generation diversity. Consequently, under repeated sampling, the model may solve certain queries more frequently, while for other queries, its probability mass may collapse onto erroneous reasoning paths. This phenomenon is closely related to entropy collapse, where the entropy of the policy decreases rapidly as training progresses (Cui et al., 2025; Yu et al., 2026).

Formally, we define the token-level policy entropy at position $k$ as the uncertainty of the policy distribution after observing the preceding tokens:

$$H_\theta(t_k \mid t_{<k}) = -\sum_{t_w\in\mathcal{V}} \pi_\theta(t_w \mid t_{<k}) \log \pi_\theta(t_w \mid t_{<k}),$$

where $\mathcal{V}$ denotes the vocabulary of the policy model. When the policy becomes highly confident in a particular token $t_w$, i.e., when $\pi_\theta(t_w \mid t_{<k})$ approaches 1, the entropy approaches 0, indicating reduced uncertainty and weaker exploration.

To enable a more fine-grained analysis of entropy dynamics, we further segment each reasoning trajectory into a sequence of reasoning steps according to newline characters and periods. Specifically, we denote a reasoning trajectory as $r = \{s_1, \cdots, s_m\}$, where each step $s_j = \{t_1, \cdots, t_l\}$ consists of a sequence of tokens. We then define the step-level entropy as the average token-level entropy within that step:

$$\bar{H}_\theta(s_j) = \frac{1}{l}\sum_{k=1}^{l} H_\theta(t_k \mid s_{<j}, t_{<k}), \tag{2}$$

where $s_{<j}$ denotes the preceding reasoning steps and $t_{<k}$ denotes the preceding tokens within the current step. Here, $\theta$ refers to the parameters of the target policy model being trained, rather than those of the auxiliary small model used to generate prefix trajectories.

## 3 Motivation: Cross-Model Prefixes Increase Policy Uncertainty

To understand how reasoning trajectories generated by weaker auxiliary models can influence the target model, we first empirically investigate the entropy discrepancy induced by cross-model prefixes.

We observe that when the target model is conditioned on token or step sequences generated by another model, it exhibits weaker adaptability to autonomous continuation than when it is conditioned on its own generated content. In particular, the target model becomes more uncertain when selecting subsequent tokens, which is reflected by higher policy entropy. This suggests that cross-model prefixes may shift the target model away from its familiar generation distribution and expose it to under-explored regions of the reasoning space.

To illustrate this phenomenon, we present a representative example in which we compute the step-level entropy of Qwen2.5-7B before and after GRPO training. We compare two types of prefixes: prefixes generated by Gemma-2-2B and prefixes generated by Qwen2.5-7B itself. The results are shown in Figure 1. In the initial reasoning steps, Qwen2.5-7B exhibits substantially higher entropy when conditioned on the Gemma-2-2B-generated prefix than when conditioned on its own generated prefix. This indicates that the target model is less certain about how to continue from reasoning states produced by the smaller model.

As generation proceeds, the step-level entropy gradually decreases and eventually approaches a similar level across the two types of prefixes. This suggests that the target model can adapt to the unfamiliar prefix after several reasoning steps, but the early-stage uncertainty provides an opportunity to broaden the exploration space. When the same analysis is conducted using the GRPO-trained model, the entropy values decrease overall, consistent with the entropy-collapse behavior of RLVR. Nevertheless, the relative trend remains similar: cross-model prefixes still induce higher uncertainty in the early reasoning steps.

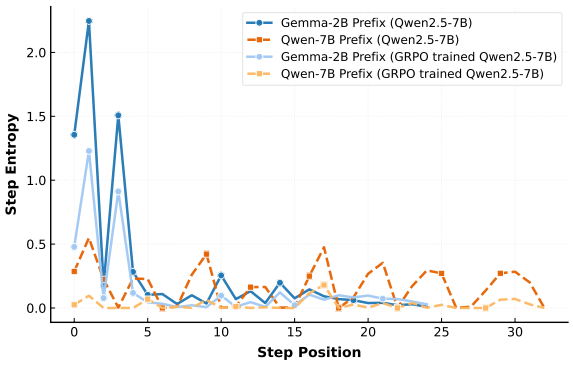



Figure 1: Step-level entropy dynamics of Qwen2.5-7B under prefixes generated by Gemma-2-2B and Qwen2.5-7B, before and after GRPO. Cross-model prefixes induce higher entropy during early reasoning, suggesting broader exploration.

This observation motivates our framework. By injecting partial reasoning trajectories generated by auxiliary models into the RLVR process, we can perturb the target model away from its over-confident reasoning paths and encourage exploration of alternative trajectories. In this way, cross-model prefixes serve as non-parametric steering signals that help mitigate diversity collapse during RLVR training.

## 4 METHODOLOGY

In standard RLVR frameworks, the model is to generate a complete solution directly from the original prompt. Given a question $q \in \mathcal{D}$, the model samples a reasoning trajectory $r$ and is optimized according to the verifier reward:

$$\theta^* = \underset{\theta}{\operatorname{argmax}}\, \mathbb{E}_{q\sim\mathcal{D},\, r\sim\pi_\theta(\cdot|q)}\big[R(q,r)\big]. \tag{3}$$

However, since each reasoning step is conditioned only on the model's own previous generations, RLVR may reinforce a limited set of high-reward trajectories and leave alternative reasoning paths under-explored.

### 4.1 PREFIX-COMPLETION RLVR

To broaden exploration, we introduce prefix-completion RLVR. Instead of always generating a full solution from scratch, the target model is conditioned on a partial reasoning trajectory generated by an auxiliary model and trained to complete the remaining solution. Formally, let

$$\tilde{r} = \{\tilde{s}_1, \cdots, \tilde{s}_L\}$$

denote an auxiliary prefix of $L$ reasoning steps, and let $r_{\text{suf}}$ denote the continuation generated by the target model. The objective becomes

$$\theta^* = \underset{\theta}{\operatorname{argmax}}\, \mathbb{E}_{q\sim\mathcal{D},\, r_{\text{suf}}\sim\pi_\theta(\cdot|q,\tilde{r})}\big[R(q,\tilde{r}\circ r_{\text{suf}})\big], \tag{4}$$

where $\circ$ denotes concatenation.

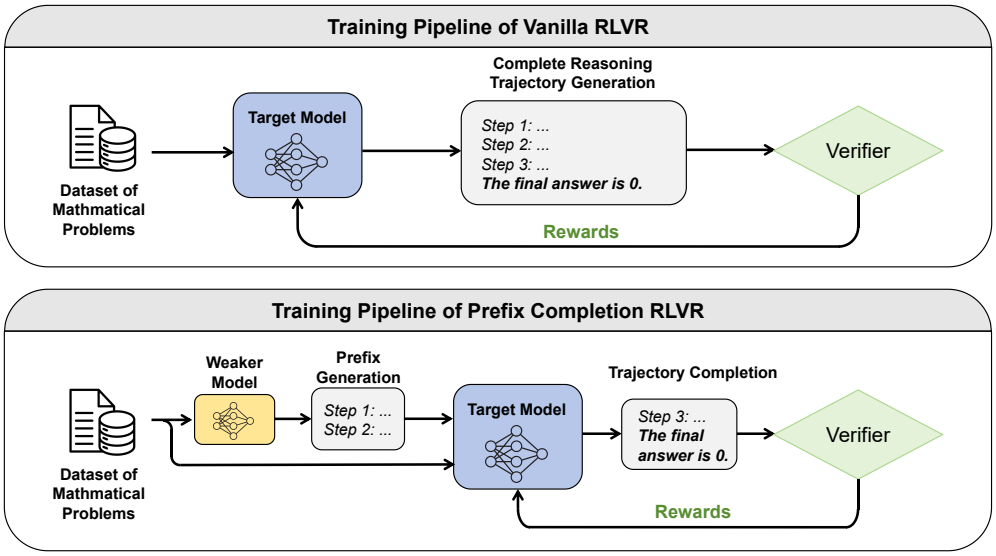



Figure 2: Overview of the proposed prefix-completion RLVR framework. Auxiliary small models first generate candidate reasoning trajectories, which are then truncated into partial prefixes. During RLVR training, the target model is trained to complete the remaining reasoning process conditioned on either the original question or the prefix-augmented input.

By completing unfamiliar prefixes, the target model is encouraged to explore reasoning states outside its dominant generation distribution. Once a successful continuation receives a positive reward, RLVR can increase the likelihood of such alternative trajectories, thereby improving reasoning coverage after training.

### 4.2 Entropy-Based Prefix Truncation

To construct prefix trajectories, we use auxiliary small models to generate complete solutions and truncate them into partial reasoning prefixes. An effective prefix should change the target model's conditioning context while leaving subsequent steps for the target model to generate. Very long prefixes may leave little reasoning for the target model to complete.

As illustrated in Section 3, cross-model prefixes can produce higher target-model entropy in the initial reasoning steps, followed by lower entropy in later steps. We use the largest adjacent decline as an operational change-point in this measured uncertainty, rather than treating it as a proven boundary between distinct reasoning regimes.

Specifically, after segmenting an auxiliary solution into $m$ reasoning steps (discarding and regenerating solutions with $m < 2$), we compute the step-level entropy using the target base model. Let

$$H_L = \bar{H}_{\theta_0}(\tilde{s}_L),$$

where $\theta_0$ denotes the parameters of the target base model before RLVR training, and $\bar{H}_{\theta_0}(\tilde{s}_L)$ is the average token-level entropy assigned along the $L$-th auxiliary reasoning step, as defined in Eq. equation 2. We select the largest observed decrease between adjacent steps:

$$L^* = \arg\max_{1 \le L < m} [H_L - H_{L+1}]. \tag{5}$$

We retain the auxiliary prefix before this decrease:

$$\tilde{r} = \{\tilde{s}_1, \ldots, \tilde{s}_{L^*}\}. \tag{6}$$

The target model subsequently generates the remaining reasoning trajectory conditioned on this truncated prefix.

The step average in Eq. equation 2 accounts for differences in step length. The entropy is measured along the given auxiliary tokens; it does not directly measure the diversity or correctness of future

continuations. Selecting the prefix just before the largest decline is a reproducible, label-free change-point heuristic that leaves the subsequent reasoning to the target model. It is not a guarantee of useful exploration or an optimal truncation point.

We compute the truncation point once using the target base model $\theta_0$ and pre-generate the prefixes before RLVR training. This preparation adds no entropy-measurement step to the training loop.

### 4.3 Mixed Training

Prefix-completion training promotes exploration, but the target model must still solve problems directly from the original question at inference time. We therefore adopt a mixed training strategy that combines standard question-only RLVR with prefix-completion RLVR.

For each training question $q$, we choose the input condition at the question level: with probability $p$, we append its pre-generated, truncated auxiliary prefix $\tilde{r}$ and train the model to complete the remaining reasoning process; with probability $1 - p$, we use the original question alone. All eight GRPO rollouts for a question share the same selected input. Prefix tokens are part of the conditioning context but are excluded from the policy loss; the verifier scores the concatenation of the prefix and generated continuation. The mixed objective is

$$\theta^* = \arg\max_{\theta} \mathbb{E}_{q\sim\mathcal{D}}\Big[(1-p)\mathbb{E}_{r\sim\pi_\theta(\cdot|q)}R(q,r) + p\mathbb{E}_{r_{\mathrm{suf}}\sim\pi_\theta(\cdot|q,\tilde{r})}R(q,\tilde{r}\circ r_{\mathrm{suf}})\Big].$$

In our main experiments, we set $p = 0.2$, balancing exploration from prefix-completion examples with the model's ability to solve problems without external prefixes.

## 5 Experiments

### 5.1 Training Setup

**Model Selection** We select small models of different model families from the target model, so that it's capable of generating solutions with diverse styles or different reasoning approaches, meanwhile not costing much computational resources. We choose Qwen2.5-7B (Team, 2024b) and Qwen2.5-Math-7B (Yang et al., 2024a) as the large models to train, and LLaMA-3.2-1B (Grattafiori et al., 2024) and Gemma-2-2B (Team, 2024a) as the small models to generate prefix.

**Dataset and Evaluation** For training, we employ GRPO algorithm (Shao et al., 2024). We utilize MATH training set (Hendrycks et al., 2021), which comprises 7500 unique problem-answer pairs. We test our model on various mathematical reasoning benchmarks, involving AIME 2024, AIME 2025, AMC 2023, MATH 500 (Hendrycks et al., 2021; Lightman et al., 2024), Minerva (Lewkowycz et al., 2022) and Olympiad Bench (He et al., 2024), and compute the average pass@$k$ across the benchmarks. We adopt an unbiased estimation method to calculate pass@$k$, with pass@$k = 1 - \binom{N-c}{k}/\binom{N}{k}$, where $N$ denotes the total number of responses sampled per question, while $c$ denotes the number of correct ones. For comparison, we also include two baselines that directly optimize pass@$k$: Pass@k Optimization (Chen et al., 2025) and SimKO (Peng et al., 2025).

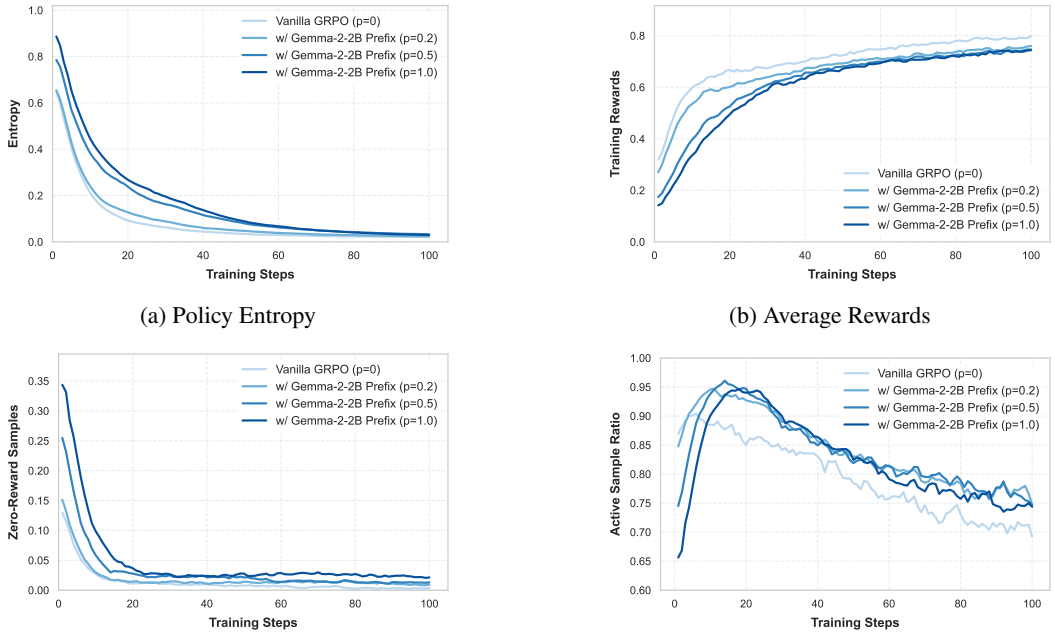
### 5.2 Main Results

The main results are summarized in Table 1. Prefix guidance provides its clearest benefit at larger values of $k$. For Qwen2.5-7B, Gemma-2-2B prefixes increase average pass@128 from 67.76 to 70.71 over vanilla GRPO, while pass@1 increases from 38.29 to 39.01. LLaMA-3.2-1B prefixes increase pass@128 from 67.76 to 69.06, but slightly reduce performance at smaller values of $k$. For Qwen2.5-Math-7B, Gemma-2-2B prefixes increase pass@128 from 69.30 to 71.22 over vanilla GRPO, while remaining below SimKO's 74.41. These results suggest a benefit for large-$k$ coverage rather than a uniform improvement in single-sample accuracy.

The per-benchmark results in Appendix D are mixed, indicating that the benefit depends on the target model and evaluation distribution. The larger average gains at higher $k$ are consistent with the

Table 1: Average pass@$k$ performance across all 6 mathematical reasoning benchmarks of different methods.

| Method | pass@1 | pass@2 | pass@4 | pass@8 | pass@16 | pass@32 | pass@64 | pass@128 |
|---|---|---|---|---|---|---|---|---|
| Qwen2.5-7B | 27.83 | 36.37 | 44.07 | 51.01 | 57.33 | 62.99 | 67.98 | 72.15 |
| + GRPO | 38.29 | 43.97 | 49.20 | 53.90 | 58.06 | 61.75 | 65.02 | 67.76 |
| + Pass@k Opt. | 35.65 | 41.12 | 46.22 | 51.03 | 55.62 | 59.78 | 63.53 | 67.12 |
| + SimKO | 38.27 | 44.17 | 49.19 | 53.40 | 57.06 | 60.62 | 64.34 | 68.28 |
| + LLaMA-3.2-1B | 38.15 -0.14 | 43.53 -0.44 | 48.52 -0.68 | 53.11 -0.79 | 57.48 -0.58 | 61.75 0.00 | 65.75 +0.73 | 69.06 +1.30 |
| + Gemma-2-2B | 39.01 +0.72 | 44.60 +0.63 | 49.52 +0.32 | 54.03 +0.13 | 58.41 +0.35 | 62.66 +0.91 | 66.78 +1.76 | 70.71 +2.95 |
| Qwen2.5-Math-7B | 26.45 | 36.83 | 46.22 | 53.77 | 59.58 | 64.29 | 68.52 | 72.71 |
| + GRPO | 40.08 | 45.86 | 51.13 | 55.91 | 60.19 | 63.72 | 66.63 | 69.30 |
| + Pass@k Opt. | 40.24 | 46.46 | 51.73 | 56.36 | 60.70 | 64.75 | 68.27 | 71.17 |
| + SimKO | 41.64 | 48.56 | 54.54 | 59.59 | 63.75 | 67.38 | 70.85 | 74.41 |
| + Gemma-2-2B | 40.11 +0.03 | 45.98 +0.12 | 51.32 +0.19 | 56.35 +0.44 | 61.04 +0.85 | 65.06 +1.34 | 68.26 +1.63 | 71.22 +1.92 |




(a) Policy Entropy

(b) Average Rewards

(c) Zero-reward Ratio

(d) Non-trivial Sample Ratio

Figure 3: Training dynamics under different prefix probabilities $p$.

intended role of cross-model prefixes as an exploration intervention, although pass@$k$ alone does not establish that the model generates more distinct reasoning paths.

## 6 Analysis

### 6.1 Training Dynamics

Figure 3 compares the training dynamics of Qwen2.5-7B with/without Gemma-2-2B prefixes.

**Dynamic Exploration via Distributional Perturbation.** As shown in Figure 3a, the prefix-guided runs exhibit higher policy entropy than the vanilla GRPO baseline, especially early in training. Increasing the prefix injection probability is also associated with higher measured entropy. This is consistent with heterogeneous prefixes exposing the target model to less familiar reasoning states.

Although the entropy eventually converges toward a level similar to the baseline, the early high-entropy phase may provide more opportunities to sample alternative reasoning paths before the policy becomes concentrated. This training pattern is consistent with delayed policy concentration under prefix conditioning; it does not establish that higher entropy causes the downstream performance gains.

**Exploration Cost and Informative Reward Signals.** The lower average reward of prefix-guided models in Figure 3b can be interpreted as the cost of exploration. Since the prefixes are generated by smaller auxiliary models, they are not always aligned with optimal solution paths and may even introduce misleading intermediate steps. However, as shown in Figure 3d, introducing these external prefixes effectively mitigates the reward signal sparsity inherent in GRPO.

In vanilla GRPO, samples within the same group often receive all-correct or all-incorrect rewards, yielding negligible relative advantage signals. With external prefixes, the observed reward patterns become more heterogeneous, increasing the proportion of groups with non-trivial relative advantages during training. Figure 3c also shows that the zero-reward ratio decreases after an initially difficult phase of conditioning on cross-model prefixes. These observations are consistent with the target model learning to complete some imperfect prefixes successfully.

### 6.2 Prefix Quality Assessment

To understand the underlying mechanism of our framework, we conducted an assessment on the prefix quality to see whether the initial reasoning steps provide inspiring or misleading guidance.

We leverage DeepSeek-V4-Flash (DeepSeek-AI, 2026) to measure the quality of the truncated prefixes generated by Gemma-2-2B, LLaMA-3.2-1B, Qwen2.5-1.5B and Qwen2.5-7B, respectively, and divide them into the following classes: (a) **No substantial guidance** (b) **Completely incorrect and misleading** (c) **Flawed but acceptable guidance** (d) **Completely correct with appropriate guidance**. We randomly pick 200 entries generated by each model, and employ DeepSeek to evaluate the prefixes based on a provided prompt template together with the original problems and ground truth answers.

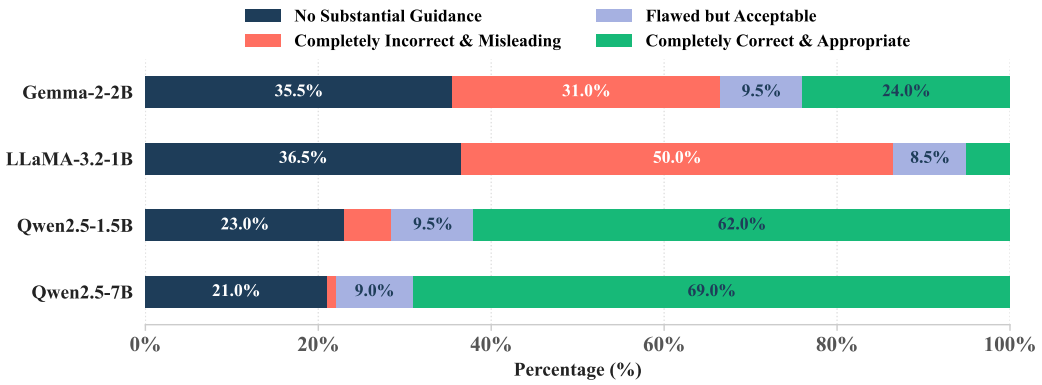



Figure 4: Fine-grained quality distribution of prefixes across different generator models.

The results are illustrated in Figure 4. Although the large-$k$ gains occur with prefix-completion training, **the majority of assessed prefixes from small models provide either no substantial guidance or incorrect and misleading directions**. This suggests that high-fidelity semantic guidance is not necessary for the observed gains in these settings.

Together with the declining zero-reward ratio during training, this result is consistent with the target model learning to complete some low-quality prefixes successfully. It also weakens a pure teacher-quality explanation for the observed large-$k$ gains, although the present analysis cannot isolate prefix correctness from distributional mismatch.

Among the non-homologous auxiliary models, Gemma-2-2B produces higher-quality prefixes than LLaMA-3.2-1B and yields larger average gains for Qwen2.5-7B. Prefixes from Qwen2.5-1.5B and

Qwen2.5-7B are assessed as higher quality but do not yield comparable pass@128 gains in the ablation. One possible explanation is that homologous Qwen models generate trajectories closer to those of the target model; this distributional explanation remains a hypothesis.

Taken together, these observations suggest that semantic guidance quality alone does not explain the differences between prefix sources. Cross-model prefixes may also act as structured trajectory-level perturbations. The current experiments do not determine how much each factor contributes.

### 6.3 Ablation Study

We examine the sensitivity of Qwen2.5-7B to the truncation rule, prefix injection probability $p$, and prefix source. Table 2 summarizes these comparisons. They help assess the proposed design choices, but do not independently identify a causal exploration mechanism.

First, entropy-based truncation modestly outperforms random step-level truncation in this setting, reaching 70.71 rather than 70.19 pass@128. The comparison evaluates a fixed, label-free rule based on the largest measured step-entropy decline. It provides limited directional support for this choice, but neither establishes its optimality nor shows that the selected point maximizes the diversity of subsequent continuations.

Second, varying the injection probability $p$ reveals a trade-off in the aggregate results: $p = 0.5$ gives the highest pass@1, whereas $p = 0.2$ gives the highest pass@128. Training exclusively on prefix-conditioned prompts ($p = 1.0$) yields lower aggregate scores than $p = 0.2$, possibly because evaluation uses question-only prompts.

Lastly, prefixes from homologous Qwen models yield no consistent gain over vanilla GRPO. Together with the prefix-quality analysis, this pattern is consistent with a role for distributional mismatch, but it does not establish trajectory perturbation as the causal mechanism.

Table 2: Ablation results on Qwen2.5-7B.

| **Method** | **pass@1** | **pass@128** |
|---|---|---|
| $p = 0$ (Vanilla GRPO) | 38.29 | 67.76 |
| *Prefix Probability p* | | |
| $p = 0.2$ | 39.01 | **70.71** |
| $p = 0.5$ | **39.22** | 67.89 |
| $p = 1.0$ | 38.56 | 68.54 |
| *Truncation Strategy* | | |
| random step truncation | 38.08 | 70.19 |
| *Homologous Prefix* | | |
| Qwen2.5-1.5B prefix | 38.88 | 67.22 |
| Qwen2.5-7B prefix | 38.12 | 68.79 |

## 7 Conclusions

We study whether partial solutions generated by weaker models can serve as trajectory-level perturbations during RLVR. Across the tested settings, their clearest benefit appears at larger values of $k$, suggesting improved reasoning coverage, while pass@1 and per-benchmark results are mixed. Training-dynamics and prefix-quality analyses are consistent with the hypothesis that distributionally mismatched prefixes encourage exploration even when their semantic guidance is imperfect. They do not, however, establish a causal relationship between entropy and downstream performance. These findings provide a complementary perspective on using small models during RLVR training.

Because the method operates at the input-trajectory level, future work could test whether weak-model prefix guidance complements objective-level RLVR regularization.

### AI use statement

In this work, we used generative AI tools for writing polishing, code refactoring, and prefix quality evaluation. We have not used generative AI tools for core algorithmic design, mathematical formulation, or primary result interpretation. The guidance quality of truncated prefixes was classified by DeepSeek-V4-Flash using the prompt provided in Appendix C; these classifications were not independently verified by human annotators. LLM-assisted code was tested for correctness and reproducibility. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

### Ethics statement

This work primarily focuses on improving the generative diversity and exploration efficiency of large language models in mathematical reasoning tasks. All experiments were conducted using publicly available benchmarks and datasets. The study does not involve human participants, sensitive personal data, or crowdsourced annotations. We do not anticipate any direct negative social impacts, ethical risks, or privacy concerns arising from this research.

### Reproducibility statement

We report the main training and evaluation settings, including the learning rate, batch size, sampling temperatures, and prefix injection probability $p = 0.2$, in Section 5.1 and the Implementation Details appendix. Prompt templates for the 7B target models, auxiliary small models, and prefix quality assessment are provided in Appendix C. The experiments use the open-source VeRL and vLLM frameworks and publicly available models and datasets.

## References

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024.

Zhipeng Chen, Xiaobo Qin, Youbin Wu, Yue Ling, Qinghao Ye, Wayne Xin Zhao, and Guang Shi. Pass@ k training for adaptively balancing exploration and exploitation of large reasoning models. *arXiv preprint arXiv:2508.10751*, 2025.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, et al. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025.

DeepSeek-AI. Deepseek-v4: Towards highly efficient million-token context intelligence, 2026.

Wenlong Deng, Yi Ren, Yushu Li, Boying Gong, Danica J Sutherland, Xiaoxiao Li, and Christos Thrampoulidis. Token hidden reward: Steering exploration-exploitation in group relative deep reinforcement learning. *arXiv preprint arXiv:2510.03669*, 2025.

Yihong Dong, Xue Jiang, Yongding Tao, Huanyu Liu, Kechi Zhang, Lili Mou, Rongyu Cao, Yingwei Ma, Jue Chen, Binhua Li, et al. Rl-plus: Countering capability boundary collapse of llms in reinforcement learning with hybrid-policy optimization. *arXiv preprint arXiv:2508.00222*, 2025.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

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 Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. 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 (Volume 1: Long Papers)*, pp. 3828–3850, 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. *NeurIPS*, 2021.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. Decomposed prompting: A modular approach for solving complex tasks. *arXiv preprint arXiv:2210.02406*, 2022.

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 ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857, 2022.

Jiazheng Li, Hongzhou Lin, Hong Lu, Kaiyue Wen, Zaiwen Yang, Jiaxuan Gao, Yi Wu, and Jingzhao Zhang. Questa: Expanding reasoning capacity in llms via question augmentation. *arXiv preprint arXiv:2507.13266*, 2025.

Xiao Liang, Zhongzhi Li, Yeyun Gong, Yelong Shen, Ying Nian Wu, Zhijiang Guo, and Weizhu Chen. Beyond pass@ 1: Self-play with variational problem synthesis sustains rlvr. *arXiv preprint arXiv:2508.14029*, 2025.

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 *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024.

Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3967–3976, 2019.

Ruotian Peng, Yi Ren, Zhouliang Yu, Weiyang Liu, and Yandong Wen. Simko: Simple pass@ k policy optimization. *arXiv preprint arXiv:2510.14807*, 2025.

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

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

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

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Stanford alpaca: An instruction-following llama model, 2023.

Gemma Team. Gemma. 2024a. doi: 10.34740/KAGGLE/M/3301. URL `https://www.kaggle.com/m/3301`.

Qwen Team. Qwen2.5: A party of foundation models, September 2024b. URL `https://qwenlm.github.io/blog/qwen2.5/`.

Christian Walder and Deep Tejas Karkhanis. Pass@ k policy optimization: Solving harder reinforcement learning problems. *Advances in Neural Information Processing Systems*, 38:152416–152445, 2026.

Fang Wu, Weihao Xuan, Ximing Lu, Mingjie Liu, Yi Dong, Zaid Harchaoui, and Yejin Choi. The invisible leash: Why rlvr may or may not escape its origin. *arXiv preprint arXiv:2507.14843*, 2025a.

Jinyang Wu, Mingkuan Feng, Shuai Zhang, Feihu Che, Zengqi Wen, Chonghua Liao, and Jianhua Tao. Beyond examples: High-level automated reasoning paradigm in in-context learning via mcts. *arXiv preprint arXiv:2411.18478*, 2024.

Jinyang Wu, Chonghua Liao, Mingkuan Feng, Shuai Zhang, Zhengqi Wen, Pengpeng Shao, Huazhe Xu, and Jianhua Tao. Thought-augmented policy optimization: Bridging external guidance and internal capabilities. *arXiv preprint arXiv:2505.15692*, 1(8):10, 2025b.

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. Learning to reason under off-policy guidance. *Advances in Neural Information Processing Systems*, 38:117157–117186, 2026.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024a.

Ling Yang, Zhaochen Yu, Tianjun Zhang, Shiyi Cao, Minkai Xu, Wentao Zhang, Joseph E Gonzalez, and Bin Cui. Buffer of thoughts: Thought-augmented reasoning with large language models. *Advances in Neural Information Processing Systems*, 37:113519–113544, 2024b.

Ling Yang, Zhaochen Yu, Bin Cui, and Mengdi Wang. Reasonflux: Hierarchical llm reasoning via scaling thought templates. *arXiv preprint arXiv:2502.06772*, 2025.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

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

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *arXiv preprint arXiv:2503.18892*, 2025.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*, 2022.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*, 2022.

# A Related Works

## A.1 Reasoning with High-level Thought Guidance

Standard methodologies utilize task decomposition (Zhou et al., 2022; Khot et al., 2022) or exemplar selection (Zhang et al., 2022) to reduce problem complexity, but these strategies rely heavily on manual engineering and primarily enhance frozen models during inference (Yang et al., 2024b; Wu et al., 2024; Yang et al., 2025). Effectively integrating abstract problem-solving guidance into on-policy RL training remains under-explored. While Yan et al. (2026) introduces off-policy guidance to stabilize learning, it depends on strict distillation signals from a strong teacher where the capability gap often exacerbates training instability. In contrast, Wu et al. (2025b) integrates external high-level thought patterns directly into RL training, enhancing intrinsic reasoning and training stability without strong-policy dependencies.

### A.2 Exploration Mechanism and Learning Dynamics

To resolve the exploration deficit in RLVR (Yue et al., 2025; Wu et al., 2025a), current strategies manipulate inputs or objectives. Data-centric methods utilize off-policy distillation (Dong et al., 2025; Li et al., 2025) or prompt paraphrasing (Liang et al., 2025) to trigger diverse trajectories, while reward-centric methods optimize group-wise rewards to directly penalize redundancy (Walder & Karkhanis, 2026; Chen et al., 2025). Alternatively, entropy-centric methods employ policy entropy (Cui et al., 2025) or bounded token confidence via Token-Hidden Reward (THR) (Deng et al., 2025) to control exploration, though THR incurs steep computational overhead. Additionally, Peng et al. (2025) attempts to mitigate policy collapse via asymmetric gradient redistribution. Our prefix-completion framework leaves the GRPO objective unchanged and uses input-level trajectory perturbations. Auxiliary prefix generation and target-model entropy measurement require additional preparation, so we do not claim that the method has no computational overhead. This data-level intervention is complementary to objective-level regularization.

Our method is an input-level training intervention rather than a new policy optimization algorithm. Unlike inference-time thought guidance, the auxiliary prefixes are used only during RLVR training; evaluation uses the original question alone. Unlike approaches that depend on a strong teacher's correct reasoning, the auxiliary models here are weaker, and many of their assessed prefixes provide little reliable guidance. We examine whether such imperfect cross-model trajectories can partially recover large-$k$ coverage lost under GRPO. The present results do not rule out contributions from longer or differently phrased training inputs.

## B Implementation Details

We implement our training framework based on VeRL (Sheng et al., 2025). All the models are trained with learning rate of $10^{-6}$ and no warm-up steps. The training batch size is set to 1024 and PPO mini batch size is set to 256. Each input question is sampled with 8 responses with temperature 1.0. KL penalty is not employed in GRPO following Zeng et al. (2025). For evaluation, we utilize vLLM (Kwon et al., 2023) to accelerate generation. The temperature is set to 0.6, top p 0.95, with max generation length 4096. For prefix generation, we set the small-model temperature to 0.4 and pre-generate one prefix per training question. Auxiliary solutions with fewer than two reasoning steps are discarded and regenerated. We filter out prefixes containing final answers to avoid leakage. For pass@$k$ calculation, we sample 128 responses per problem for MATH 500, Minerva and Olympiad Bench, and 200 responses for AIME 2024, AIME 2025 and AMC 2023 which are relatively small.

## C Prompt Templates

We provide the prompt templates used for training and evaluation in our experiments. For 7B models namely Qwen2.5-7B and Qwen2.5-Math-7B, we are using the original chat template using SimpleRL (Zeng et al., 2025) style prompt in Figure 5. For other smaller models involving Gemma-2-2B, LLaMA-3.2-1B and Qwen2.5-1.5B, we use prompt illustrated in Figure 6 for high-quality prefix generation. For prefix quality assessment, we adopt a prompt template shown in Figure 7.

**Prompt For 7B Models**

<—im_start—>system
You are a helpful assistant.<—im_end—>
<—im_start—>user
{question}
Please reason step by step, and put your final answer within \boxed{}. <—im_end—>
<—im_start—>assistant

Figure 5: Prompt for 7B models

**Prompt For Smaller Models**

You are a helpful assistant.

Question:

{question}

Solution:

Figure 6: Prompt for smaller models

## D PER-BENCHMARK RESULTS

To provide a more fine-grained evaluation, we report the per-benchmark results across all six mathematical reasoning benchmarks in Tables 3. We include the same set of baselines and model variants as in the main experiments, and report their pass@1/pass@128 performance separately on each benchmark.

Table 3: Per-benchmark results formatted as **pass@1 / pass@128**.

| Method | AIME24 | AIME25 | AMC23 | MATH-500 | Minerva | OlympiadBench |
|---|---|---|---|---|---|---|
| Qwen2.5-7B | 7.62 / 54.57 | 3.90 / 45.01 | 36.95 / 96.92 | 62.88 / 96.40 | 26.29 / 65.07 | 29.35 / 74.93 |
| + GRPO | 17.48 / 49.58 | 7.98 / 41.50 | 50.98 / 92.12 | 76.52 / 92.80 | 37.44 / 59.93 | 39.37 / 70.62 |
| + Pass@k Opt. | 14.87 / 49.08 | 6.55 / 42.62 | 48.04 / 92.82 | 72.63 / 94.00 | 35.67 / 55.88 | 36.16 / 68.30 |
| + SimKO | 14.00 / 53.40 | 7.00 / 41.40 | 56.29 / 89.36 | 76.40 / 93.20 | 36.11 / 61.40 | 39.82 / 70.92 |
| + LLaMA-3.2-1B | 15.43 / 56.81 | 6.50 / 40.50 | 55.81 / 96.04 | 75.89 / 93.40 | 35.14 / 57.72 | 40.14 / 69.88 |
| + Gemma-2-2B | 18.62 / 58.50 | 7.80 / 43.98 | 53.97 / 96.81 | 76.64 / 93.80 | 36.19 / 57.72 | 40.84 / 73.44 |
| Qwen2.5-Math-7B | 13.35 / 66.01 | 5.38 / 39.63 | 38.25 / 96.87 | 55.67 / 96.20 | 19.48 / 61.76 | 26.54 / 75.82 |
| + GRPO | 23.58 / 57.43 | 8.92 / 40.86 | 55.81 / 88.59 | 77.48 / 93.60 | 34.60 / 63.24 | 40.07 / 72.11 |
| + Pass@k Opt. | 22.15 / 62.63 | 10.27 / 41.68 | 59.94 / 96.22 | 75.00 / 95.60 | 35.56 / 58.46 | 38.54 / 72.44 |
| + SimKO | 27.53 / 68.48 | 10.73 / 52.06 | 58.96 / 96.01 | 76.92 / 95.80 | 36.11 / 61.40 | 39.57 / 72.74 |
| + Gemma-2-2B | 21.72 / 65.49 | 10.53 / 45.97 | 59.80 / 91.52 | 76.05 / 92.60 | 32.81 / 59.93 | 39.74 / 71.81 |

## Prompt For Prefix Assessment

**System:**
You are an expert mathematical evaluator. Your task is to analyze an incomplete reasoning prefix generated by a small model and evaluate the quality of guidance it provides for a target model to complete the problem. You must output your response strictly as a valid JSON object.

Classification Criteria:
1. 'Completely Correct & Appropriate Guidance': The prefix logic and mathematical direction are entirely correct. It provides an optimal, flawless foundation that directly guides the next steps toward the correct answer.
2. 'Flawed but Acceptable Guidance': The prefix contains minor computational slips, notation errors, or a slightly suboptimal route, but the core direction is reasonable and salvageable. It still provides a passable foundation.
3. 'No Substantial Guidance': The prefix consists of trivial statements, repetitive filler text, or re-stating the prompt without making any real progress. It provides neither helpful direction nor harmful errors.
4. 'Completely Incorrect & Misleading Guidance': The prefix contains severe mathematical violations, fundamental logical collapses, or deep hallucinations. It actively misleads the reasoning path towards a dead-end or an incorrect answer.

**User:**
<input_data>
[Mathematical Problem]:
{question}

[Reference Answer & Final Result]:
{ground_truth_answer}

[Small Model Reasoning Prefix (Incomplete Segment)]:
{small_model_prefix}
</input_data>

<output_format>
Return a JSON object with the exact following keys:
{{
"category": "One of the four categories defined in the system prompt (exactly as stated).",
"prefix_logic_analysis": "A concise 1-2 sentence mathematical analysis of the prefix text (strictly under 60 words).",
"guidance_impact_justification": "Detailed explanation of how this specific level of guidance affects the downstream completion (e.g., whether it accelerates correct derivation, requires error-correction, wastes tokens, or derails the model)."
}}
</output_format>

Figure 7: Prompt for DeepSeek-v4-Flash prefix assessment.