REASONING OFF THE BEATEN PATH: EXPANDING SOLUTION COVERAGE IN RLVR VIA PREFIX TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reasoning diversity is crucial for large language models: exploring multiple high-quality reasoning paths broadens solution coverage at inference time and increases the chance of discovering high-reward trajectories during reinforcement learning with verifiable rewards (RLVR). However, a model's reasoning distribution often concentrates on a few solution modes, and RL optimization can further exacerbate this tendency, making exploration increasingly ineffective. The challenge is thus to promote structured reasoning diversity without sacrificing solution quality. To this end, we propose Information-Maximizing Augmented eXploration (IMAX), a framework that learns a lightweight pool of soft prefixes to elicit diverse reasoning paths from a frozen language model. IMAX combines a verifiable correctness reward to steer each prefix toward successful reasoning with a response-level Information Maximization (InfoMax) reward that prevents different prefixes from inducing redundant behaviors. Once learned, these prefixes can broaden solution coverage at inference time and diversify rollouts for subsequent full-policy RLVR. Experiments across three backbone scales and multiple mathematical reasoning benchmarks demonstrate that IMAX improves over matched prefix-tuned RLVR baselines, yielding gains of up to 11.60% in Pass@4 and 8.08% in Avg@4. Beyond these inference-time gains, the learned prefixes also enhance subsequent full-policy RLVR by promoting more effective exploration during training.

1 INTRODUCTION

Both test-time scaling and reinforcement learning with verifiable rewards (RLVR) rely on sampling high-quality candidate solutions from the model's reasoning distribution (Karan & Du, 2025; Chen et al., 2025b; Gao et al., 2025; Hu et al., 2025). Although RLVR has substantially improved mathematical reasoning (Guo et al., 2025; Yu et al., 2025), its rollout distribution can progressively concentrate on a few dominant solution modes, as reflected by the entropy collapse phenomenon observed during training (Cui et al., 2025; Wang et al., 2025). Consequently, RLVR may improve Pass@1 while failing to preserve or expand Pass@K, indicating that stronger single-rollout performance does not necessarily translate into broader solution coverage (Yue et al., 2025; Chen et al., 2025b). The central challenge is therefore not merely to inject more randomness, but to elicit diverse, task-relevant reasoning trajectories without sacrificing solution quality.

Existing approaches to improving RLVR exploration typically modify the optimization objective, introduce exploration or diversity rewards, or intervene in rollout sampling (Gao et al., 2025; Zhang et al., 2025a; Hu et al., 2025; Shah et al., 2026). These interventions nevertheless operate through a policy distribution that is strongly shaped by the base model's prior over reasoning trajectories (Wu et al., 2025; Yue et al., 2025; Karan & Du, 2025). This limitation motivates a complementary strategy: rather than encouraging more varied samples from a single reasoning prior, we explicitly construct multiple controllable reasoning priors. Prompt-based adaptation provides a lightweight mechanism for doing so, since learned prompts can steer a frozen model toward distinct behaviors by changing its conditioning context (Li & Liang, 2021; Lester et al., 2021; Liu et al., 2022; Qin & Eisner, 2021). We therefore optimize a pool of soft prefixes, each of which reshapes the rollout distribution for the same problem and exposes a different region of the model's reasoning space.

Building on this idea, we propose Information-Maximizing Augmented eXploration (IMAX), a soft-prefix framework that learns multiple controllable reasoning priors, as illustrated in Figure 1. IMAX

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

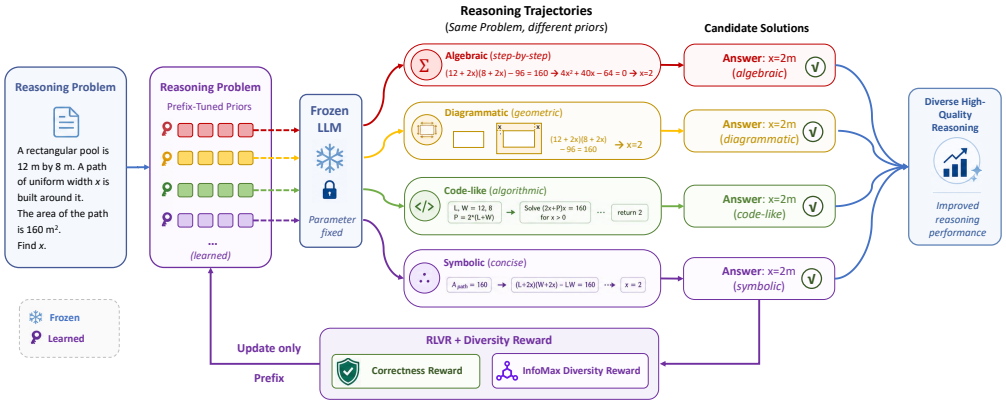


Figure 1: Overview of the IMAX framework. For a reasoning problem, a frozen LLM is conditioned on a learned soft-prefix pool, where different prefixes induce different reasoning priors and lead to diverse candidate trajectories, such as algebraic, diagrammatic, code-like, or symbolic solutions. Training updates only the prefix priors using RLVR correctness rewards and the InfoMax reward, encouraging prefixes to produce high-quality and non-redundant reasoning trajectories.

keeps the backbone LLM frozen and optimizes only a pool of soft prefixes. Each prefix is initialized from a textual prompt encoding a different reasoning strategy and serves as a lightweight conditioning handle that shifts the rollout distribution toward a distinct region of the model's reasoning space. We optimize the prefix pool with two complementary signals: the verifiable reward encourages each prior to produce correct solutions, while a response-level Information Maximization (InfoMax) reward discourages different prefixes from inducing redundant trajectories. Specifically, the InfoMax reward is derived from a variational lower bound on the conditional mutual information between the prefix identity and the generated trajectory, encouraging the induced reasoning behaviors to remain distinguishable across prefixes. Together, these objectives elicit diverse, task-relevant reasoning trajectories. In summary, our main contributions in this paper are threefold:

1. We introduce a lightweight prefix-tuned RLVR framework that learns a pool of controllable reasoning priors while keeping the backbone LLM frozen.
2. We derive a response-level InfoMax reward from a variational lower bound on the conditional mutual information between the prefix identity and the generated trajectory, encouraging different prefixes to induce distinguishable, task-relevant reasoning behaviors.
3. Experiments across three backbone scales and multiple mathematical reasoning benchmarks show that IMAX improves inference-time solution coverage and average rollout quality over matched prefix-tuned RLVR baselines. On Qwen3-8B, the learned prefixes further improve subsequent full-policy RLVR when used only during training rollouts.

2 Related Work

Reinforcement learning from verifiable rewards. RLVR has become a standard approach to improving LLM reasoning through verifiable feedback (Jaech et al., 2024; Shao et al., 2024; Guo et al., 2025; Yu et al., 2025; Zheng et al., 2025a; Tan et al., 2025). However, its policy distribution can concentrate on a narrow set of solutions, as reflected by entropy collapse and evidence that some gains arise from increasing confidence in existing reasoning modes rather than discovering new ones (Cui et al., 2025; Wang et al., 2025; Yue et al., 2025; Zhu et al., 2025).

Existing methods improve RLVR exploration by modifying the optimization objective (Yu et al., 2025; Chen et al., 2025b), adding exploration or diversity rewards (Gao et al., 2025; Zhang et al., 2025a; Hu et al., 2025), or intervening in rollout generation (Zheng et al., 2025b; Bai et al., 2026). A concurrent method, UpSkill (Shah et al., 2026), also uses mutual information to induce structured response diversity during full-policy RLVR. In contrast, IMAX freezes the backbone and learns a pool of input-level soft prompts that reshape the rollout distribution before generation; a detailed comparison is provided in Appendix A.

Soft prompting. Soft prompts provide lightweight control over models, while prompt pools have been used to elicit latent behaviors, transfer across tasks, and serve as compact control modules (Qin & Eisner, 2021; Vu et al., 2022; Petrov et al., 2024; Wang et al., 2022b;a; Smith et al., 2023). IMAX instead optimizes a pool of input-level continuous prompts with verifiable and InfoMax rewards to induce distinct reasoning trajectories, throughout the paper, *prefix* denotes such a soft prompt.

3 PRELIMINARIES

We first formulate RLVR under prefix tuning in Section 3.1, and then present GRPO for prefix tuning in Section 3.2, the main RLVR algorithm used in this paper.

3.1 RLVR FOR PREFIX TUNING

Soft prompt tuning adapts a frozen language model by optimizing continuous prompt embeddings. Given an input prompt $x = (x_1, \ldots, x_L)$, a soft prompt $e = [\mathbf{e}_1, \ldots, \mathbf{e}_m] \in \mathbb{R}^{m \times d}$ consists of m trainable virtual-token embeddings with the same hidden dimension d as the model token embeddings. The model conditions on the concatenated input-embedding sequence $[\mathbf{e}_1, \ldots, \mathbf{e}_m, \mathrm{Emb}(x_1), \ldots, \mathrm{Emb}(x_L)]$, where $\mathrm{Emb}(\cdot)$ denotes the input embedding table of the base model. Thus, the soft prompt is prepended at the input-embedding layer before the Transformer blocks, while the backbone parameters remain fixed. When initialized from a text prompt, the initialization text is tokenized and mapped through the input embedding table and the resulting vectors initialize the soft prompt and are then optimized directly as continuous parameters.

We consider RLVR for prefix tuning, where the objective is to improve reasoning performance using verifiable reward signals while updating only the soft prefixes. The backbone LLM remains frozen, and the trainable parameters are a pool $\mathcal{E} = \{e_1, \ldots, e_C\}$ of C soft prefixes, with $e_z \in \mathbb{R}^{m \times d}$. For a prefix identity z, the model generates a response $y = (y_1, \ldots, y_{L_y})$ from

$$\pi_{\mathcal{E}}(y \mid x, z) \equiv \pi_{\theta_0}(y \mid x, e_z) = \prod_{\ell=1}^{L_y} \pi_{\theta_0}(y_\ell \mid x, e_z, y_{<\ell}), \tag{1}$$

where θ_0 denotes the frozen base-LLM parameters. In RLVR, the environment provides a binary reward $R(x, y) \in \{0, 1\}$ for each prompt-response pair, indicating whether the final answer is correct. We use $p(x)$ for the training-prompt distribution and $p(z)$ for the prior over the C prefix identities. The prefix-tuning RLVR objective is to maximize the expected reward over both prefix sampling and response generation:

$$\max_{\mathcal{E}} \; \mathbb{E}_{x \sim p(x),\, z \sim p(z),\, y \sim \pi_{\mathcal{E}}(\cdot \mid x, z)} \left[R(x, y) \right]. \tag{2}$$

3.2 FORMULATION OF GRPO

GRPO is a group-relative RL algorithm specifically designed for LLMs that avoids training a separate critic by estimating advantages through *group-wise relative comparison* among sampled responses. In our prefix-tuning setting, the group is formed within each assigned prefix. Given an input prompt x and a prefix index z, GRPO samples a group of N responses $\{y^{(n)}\}_{n=1}^{N}$ under the same prefix-conditioned policy, $y^{(n)} \sim \pi_{\mathcal{E}_{\text{old}}}(\cdot \mid x, z)$, where $\mathcal{E}_{\text{old}}$ denotes the prefix pool used to collect trajectories. If multiple prefix identities are assigned to the same prompt, each prompt–prefix pair (x, z) defines its own GRPO group.

The responses are evaluated by the verifier to obtain rewards $\{R(x, y^{(n)})\}_{n=1}^{N}$, and the normalized advantage is $A^{(n)} = (R(x, y^{(n)}) - \mu_R)/\sigma_R$, where μ_R and σ_R are the group mean and standard deviation. The clipped GRPO objective is

$$\mathcal{L}_{\text{GRPO}}(\mathcal{E}) = \mathbb{E}_{x,z,n,t}\big[\min\big(\rho_t^{(n)}(\mathcal{E})A^{(n)}, \tilde{A}_t^{(n)}(\mathcal{E})\big)\big], \quad \rho_t^{(n)}(\mathcal{E}) = \frac{\pi_{\mathcal{E}}(y_t^{(n)} \mid x, z, y_{<t}^{(n)})}{\pi_{\mathcal{E}_{\text{old}}}(y_t^{(n)} \mid x, z, y_{<t}^{(n)})}. \tag{3}$$

where $\tilde{A}_t^{(n)}(\mathcal{E}) = \mathrm{clip}\big(\rho_t^{(n)}(\mathcal{E}), 1-\epsilon, 1+\epsilon\big)A^{(n)}$ is the clipped surrogate term and ϵ is a clipping hyperparameter. In the original GRPO formulation, a KL regularizer is added to limit policy deviation. As our goal is to encourage prefix-conditioned exploration, we omit this KL term, following the practice adopted in DAPO (Yu et al., 2025).

4 METHODOLOGY

In this section, we first introduce prefix-conditioned reasoning in Section 4.1, and then derive a variational lower bound of the InfoMax reward in Section 4.2 for practical implementation. The detailed post-training procedure is stated in Section 4.3.

4.1 PREFIX-CONDITIONED REASONING

We train a soft-prefix pool $\mathcal{E} = \{e_1, \ldots, e_C\}$, where each prefix index z selects a distinct rollout distribution $\pi_{\mathcal{E}}(y \mid x, z)$. We initialize the prefix pool from a text prompt pool $\mathcal{T} = \{t_1, \ldots, t_C\}$ generated by GPT-5.4 (Singh et al., 2025), with deliberate prompting on producing diverse and minimally overlapping initial prompts. The resulting default prompt pool is listed in Table 9 in Appendix D.2. Each text prompt t_z is mapped through the frozen embedding layer of the backbone model and used to initialize e_z, after which e_z is optimized as continuous parameters. This initialization gives the prefix pool semantically meaningful and diverse starting points. A training-free control in Appendix C.1 confirms that semantic prefixes can shift both the coverage and average quality of samples from a frozen model, which motivates prefix-level control through post-training.

To make the prefix pool useful, different prefixes should induce high-quality and distinguishable reasoning trajectories. To enable the former, we use RL algorithm to perform online training. The requirement of diversity is to prevent the case that learned pool may collapse to redundant prefixes, where $\pi_{\mathcal{E}}(y \mid x, z) \approx \pi_{\mathcal{E}}(y \mid x, z')$ for different z and z'. Therefore, we adopt an *information-guided principle*: the prefix identity should capture salient variations in the inherent reasoning paths conditioned on the same input. Concretely, we maximize the conditional mutual information between the prefix identity Z and the generated response Y given the prompt X, which averages their fixed-prompt dependence over $p(x)$:

$$I(Z; Y \mid X) = \mathbb{E}_{p(x)}\left[\mathbb{E}_{p(z,y|x)}\left[\log \frac{p(z, y \mid x)}{p(z \mid x)p(y \mid x)}\right]\right],$$

thereby encouraging different soft prefixes to induce distinct and valid reasoning behaviors for the same question.

Alternatively, we may write in terms of joint and conditional entropies as

$$\begin{aligned} I(Z; Y \mid X) &= H(Y \mid X) - H(Y \mid X, Z) \quad \text{// forward} \\ &= H(Z \mid X) - H(Z \mid X, Y) \quad \text{// reverse} \end{aligned} \tag{4}$$

where $H(\cdot)$ denotes entropy. In Equation (4), "forward" and "reverse" denote two equivalent decompositions. The forward form follows the generation direction $Z \to Y$, while the reverse form follows the inference direction $Y \to Z$. Intuitively, the mutual information term $I(Z; Y \mid X)$ is zero if the response variable Y is independent of the prefix variable Z given the prompt X, and increases as the prefix-conditioned rollout distributions become more distinguishable.

4.2 VARIATIONAL INFOMAX REWARD

To incorporate the InfoMax reward into prefix-pool training, we consider the mutual information in the reverse form of Equation (4) for practical computation. A detailed discussion on this choice is deferred to Appendix B. We first rewrite the conditional mutual information as

$$I(Z; Y \mid X) := H(Z \mid X) - H(Z \mid X, Y) = \mathbb{E}_{p(x)}\left[\mathbb{E}_{p(z,y|x)}\left[\log p(z \mid x, y) - \log p(z \mid x)\right]\right]. \tag{5}$$

The posterior $p(z \mid x, y)$ is generally intractable, so we introduce a variational distribution $q_\phi(z \mid x, y)$ to approximate it and derive a variational lower bound on $I(Z; Y \mid X)$. Since z indexes one of C prefix identities, q_ϕ is implemented as a C-way classifier that predicts which soft prefix generated the response. This yields the exact decomposition

$$\begin{aligned} I(Z; Y \mid X) = \mathbb{E}_{p(x)}&\left[\mathbb{E}_{p(z,y|x)}\left[\log q_\phi(z \mid x, y) - \log p(z \mid x)\right]\right] \\ &+ \mathbb{E}_{p(x)}\left[\mathbb{E}_{p(y|x)}\left[\mathrm{KL}(p(z \mid x, y) \,\|\, q_\phi(z \mid x, y))\right]\right], \end{aligned}$$

and by dropping the last non-negative KL divergence term, we can immediately have the lower bound of $I(Z; Y \mid X)$ as

$$I(Z; Y \mid X) \geq \mathbb{E}_{p(x)}\left[\mathbb{E}_{p(z,y|x)}\left[\log q_\phi(z \mid x, y) - \log p(z \mid x)\right]\right]. \tag{6}$$

The above expectation is taken with respect to the conditional distribution $p(z, y \mid x)$. We leverage a standard transform to rewrite the expectation under the tractable sampling scheme $p(z \mid x)\, \pi_{\mathcal{E}}(y \mid x, z)$. Here, z is sampled from the prefix-identity distribution, and the response is sampled as $y \sim \pi_{\mathcal{E}}(\cdot \mid x, z)$ conditioned on both the prompt and the selected soft prefix.

For the variational objective, we consider the case where the prefix prior is independent of the prompt, where $p(z \mid x) = p(z) = \mathrm{Uniform}\{1, \ldots, C\}$, which is a standard design choice also adopted in previous works (Chen et al., 2016). Consequently, the mutual information objective reduces to

$$\begin{aligned} I(Z; Y \mid X) &\geq \mathbb{E}_{\substack{x \sim p(x),\, z \sim p(z), \\ y \sim \pi_{\mathcal{E}}(\cdot \mid x, z)}} \big[\log q_\phi(z \mid x, y) - \log p(z) \big] \\ &= \mathbb{E}_{\substack{x \sim p(x),\, z \sim p(z), \\ y \sim \pi_{\mathcal{E}}(\cdot \mid x, z)}} \big[\log q_\phi(z \mid x, y) \big] + H(Z). \end{aligned} \tag{7}$$

Here $H(Z)$ is constant across all samples, and we omit it from the objective.

We then incorporate the variational lower bound in Equation (7) for conditional mutual information into the RLVR objective as the InfoMax reward, which now reads

$$\max_{\mathcal{E}} \; \mathbb{E}_{x \sim p(x), z \sim p(z), y \sim \pi_{\mathcal{E}}(\cdot \mid x, z)} \Big[R(x, y) + \beta \log q_\phi(z \mid x, y) \Big]. \tag{8}$$

A hyperparameter β controls the strength of the InfoMax reward compared with the verifiable reward. Notably, this InfoMax reward is defined at the response level: q_ϕ predicts the prefix identity from the prompt x and the complete generated response y, because the semantic reasoning path induced by a prefix is expressed by the trajectory as a whole rather than by isolated token-level deviations.

4.3 Detailed Post-training Procedure

We detail the adaptation of the above variational framework to our prefix-tuning procedure in this section. The variational posterior network $q_\phi(z \mid x, y)$ encodes the prompt–response sequence with the same frozen LLM backbone and applies a C-way classification head to the hidden state of the last non-padding token. Only this classification head is trained during update.

Alternating optimization. We update q_ϕ and the prefix pool $\mathcal{E}$ in an alternating manner for each online training batch. This alternating procedure follows the spirit of coordinate ascent variational inference (CAVI) (Blei et al., 2017). With the prefix pool fixed, updating q_ϕ tightens the variational posterior used in the InfoMax lower bound; with q_ϕ fixed, updating $\mathcal{E}$ improves the prefix-conditioned rollout distribution under the current variational reward estimate. Before updating the prefix pool, we ensure that the proxy posterior network q_ϕ is sufficiently updated within the batch so that the variational lower bound can be reliably estimated, as the InfoMax reward must be computed prior to the RLVR optimization step. To compute $q_\phi(z \mid x, y)$, we perform an additional forward pass of the frozen backbone on the original prompt–response pair (x, y) without the soft prefix, thereby preventing information leakage of z through the prefix embedding itself. When training the posterior network q_ϕ, the prefix pool is fixed. Since q_ϕ is lightweight, its training incurs negligible computational overhead. After updating q_ϕ, the prefix pool $\mathcal{E}$ is optimized with the combined verifiable reward and InfoMax reward using the RLVR algorithm.

Incorporation into RLVR paradigm. Specifically, some adaptations are required for the group-relative RL algorithms. Let N denote the number of rollouts in each GRPO group. In our implementation, we realize the uniform prefix prior through stratified assignment rather than independent sampling: every input prompt x is paired with each prefix identity $z \in \{1, \ldots, C\}$, and the corresponding continuous prefix e_z is prepended to form the prefix-conditioned input. For each prompt–prefix pair (x, z), we sample N rollouts from $\pi_{\mathcal{E}}(\cdot \mid x, z)$ and treat these N responses as one GRPO group, under which the group-relative baseline can be computed in a standard manner. This stratified assignment keeps the empirical marginal over prefix identities uniform, and yields a prompt layout that is more convenient for batched parallel generation than random sampling in modern rollout engines. The complete IMAX training procedure is summarized in Algorithm 1.

5 Experiments

We conduct extensive experiments to validate the effectiveness of the proposed IMAX framework. The experimental setup is listed in Section 5.1. The main results on reasoning benchmarks are

Algorithm 1 IMAX

Input: base RLVR algorithm $\mathcal{A}$, verifiable reward R, prompt distribution $p(x)$, initialization prompts $\mathcal{T} = \{t_c\}_{c=1}^{C}$
Hyperparameters: batch size B, group size N, total steps S, InfoMax reward weight β
Initialize: prefix pool $\mathcal{E} = \{e_c\}_{c=1}^{C}$ from $\mathcal{T}$, posterior head parameters ϕ
for $s = 1$ **to** S **do**
  Sample prompts $\{x_i\}_{i=1}^{B} \sim p(x)$
  Assign prefixes stratifiedly: pair every x_i with all identities $z_c = c$, $c = 1, \ldots, C$
  Collect rollouts $y_{i,c}^{(n)} \sim \pi_{\mathcal{E}}(\cdot \mid x_i, z_c)$, $n = 1, \ldots, N$
  Compute verifiable rewards $r_{i,c}^{(n)} \leftarrow R(x_i, y_{i,c}^{(n)})$
  Fix $\mathcal{E}$ and update q_ϕ on prompt–response pairs without the soft prefix by minimizing
  $\mathcal{L}_q(\phi) \leftarrow -\dfrac{1}{BCN} \displaystyle\sum_{i,c,n} \log q_\phi(z_c \mid x_i, y_{i,c}^{(n)})$
  Compute response-level InfoMax rewards $r_{i,c}^{\mathrm{InfoMax},(n)} \leftarrow \log q_\phi(z_c \mid x_i, y_{i,c}^{(n)})$
  Form augmented rewards $\tilde{r}_{i,c}^{(n)} \leftarrow r_{i,c}^{(n)} + \beta\, r_{i,c}^{\mathrm{InfoMax},(n)}$
  For each (i, c), keep $\{(y_{i,c}^{(n)}, \tilde{r}_{i,c}^{(n)})\}_{n=1}^{N}$ as a separate GRPO group
  Fix q_ϕ and update $\mathcal{E}$ using $\mathcal{A}$ with the grouped rollouts and rewards
end for
Output: trained prefix pool $\mathcal{E}$

presented in Section 5.2; Section 5.3 presents an ablation study on the IMAX key design choices. Finally, a detailed analysis on model behavior is provided in Section 5.4.

5.1 Experimental Setup

Models and baselines. We evaluate our method on Qwen2.5-1.5B-Instruct (Yang et al., 2024a), Qwen3-4B-Base, and Qwen3-8B-Base (Yang et al., 2025). We compare against full-policy GRPO (GRPO-full), text-prompt initialization (Text Init.), and representative RLVR baselines including GRPO (GRPO-prefix), DAPO (DAPO-prefix), DIVER (DIVER-prefix) (Hu et al., 2025), and MERCI (MERCI-prefix) (Zhang et al., 2025b) under the prefix-tuning setting. For the record, we adopt the exploration intrinsic rewards in DIVER and MERCI, and explicitly adapt them into our prefix-tuning setting. We adopt the default reward weights used in the original papers. We train the models on the MATH (Hendrycks et al., 2021) dataset, which consists of 7.5k rows of data.

Implementation details. We implement our algorithms based on the TRL framework [1], all experiments are conducted on 1 × RTX PRO 6000-96GB GPU. we set the number of prefixes assigned to each prompt to be $C = 2$, and set the GRPO group size to $N = 4$ rollouts per prompt–prefix pair. For optimization, we use AdamW with a learning rate of 5×10^{-4}. The weight for the InfoMax reward is set to be 0.01. More implementation details are reported in Appendix D.

Evaluation details. For evaluation, we use lighteval [2] to test the algorithms on mathematical reasoning benchmarks including MATH-500 (Lightman et al., 2023), GSM8K (Cobbe et al., 2021), Minerva Math (Lewkowycz et al., 2022), and AMC (Hendrycks et al., 2021), and to show that our tuned prefixes do not degrade performance on general tasks, we also evaluate on an instruction following task–IFEval (Zhou et al., 2023). For prefix-tuned methods, evaluation uses the same stratified prefix assignment paradigm as in training: each prompt is paired with distinct C learned prefixes and the same number of rollouts N_{eval} is generated for each prompt–prefix pair. We set $C = 2$, and $N_{\text{eval}} = 2$, yielding rollouts number $K = 4$ for each original prompt in evaluation.

Downstream full-policy evaluation. To test whether the learned prefixes can also improve exploration during RLVR training, we conduct a separate second-stage experiment on Qwen3-8B. We freeze the learned prefix pool and use it only to condition rollout generation while updating the full policy with GRPO. The rollout group size is fixed to eight, with four rollouts assigned to each of

[1] https://github.com/huggingface/trl.
[2] https://github.com/huggingface/lighteval.

Table 1: Inference-time results on reasoning and instruction-following benchmarks. We report percentages (%) for readability. The full table with per-example standard errors is provided in Appendix C. MATH is the abbreviation for MATH-500. Darker shading marks stronger results within each metric block, while unshaded Base and GRPO-full rows are shown as reference baselines.

		Pass@4					Avg@4				
Model	**Method**	**GSM8K**	**MATH**	**Minerva**	**AMC**	**IFEval**	**GSM8K**	**MATH**	**Minerva**	**AMC**	**IFEval**
Qwen2.5-1.5B	Base	78.92	49.60	13.97	68.22	52.31	53.43	32.85	6.99	35.28	37.75
	GRPO-full	84.84	**60.20**	18.01	68.06	50.28	**74.09**	**47.55**	**10.94**	34.45	35.26
	Text Init.	80.29	50.20	16.18	68.31	54.34	56.92	34.35	8.82	34.70	38.08
	DIVER-prefix	80.29	56.20	16.54	67.72	**54.90**	59.76	37.05	7.72	33.28	38.03
	MERCI-prefix	82.11	59.00	16.18	68.31	54.53	63.34	40.90	8.64	35.05	37.43
	GRPO-prefix	85.75	49.40	15.44	**69.22**	53.60	66.89	34.80	8.27	35.38	37.11
	GRPO+IMAX	85.75	**60.00**	**18.38**	68.06	53.05	67.40	42.30	9.47	34.45	36.78
	DAPO-prefix	**86.28**	56.40	17.28	68.97	51.94	**69.50**	41.60	9.19	**35.53**	36.74
	DAPO+IMAX	84.99	59.40	15.81	68.39	53.79	67.63	**42.50**	**9.83**	34.99	**38.12**
Qwen3-4B	Base	90.60	52.00	22.43	54.13	48.80	59.70	30.00	10.57	22.83	26.94
	GRPO-full	91.81	**69.20**	25.37	71.14	43.25	**86.64**	**59.80**	**20.13**	44.02	29.25
	Text Init.	88.55	51.20	24.26	67.39	50.65	53.49	27.50	10.02	35.22	28.97
	DIVER-prefix	91.43	57.80	23.53	66.97	52.31	66.49	31.75	9.93	35.38	30.41
	MERCI-prefix	90.83	60.80	20.96	64.80	51.20	62.51	30.75	9.56	34.92	31.47
	GRPO-prefix	93.48	57.60	21.69	70.64	51.39	71.97	43.45	13.33	**39.64**	31.93
	GRPO+IMAX	**94.47**	**69.20**	23.90	**71.06**	51.57	**76.14**	**50.45**	11.86	38.07	30.64
	DAPO-prefix	93.18	56.60	**25.74**	69.06	**56.19**	72.02	41.65	**14.52**	38.32	**35.95**
	DAPO+IMAX	93.78	65.80	23.90	64.97	55.08	72.93	42.70	10.85	35.78	33.13
Qwen3-8B	Base	88.10	58.60	18.01	41.12	48.06	55.44	33.65	8.73	18.58	27.26
	GRPO-full	91.58	**74.20**	**29.41**	**74.23**	47.87	**76.95**	**65.30**	**22.33**	**47.46**	31.24
	Text Init.	90.30	59.80	20.96	62.39	52.50	56.16	33.60	8.36	31.15	30.87
	DIVER-prefix	90.75	63.00	21.69	62.89	**55.27**	57.15	36.65	9.56	29.69	32.76
	MERCI-prefix	93.18	66.40	18.38	66.47	54.71	59.59	37.95	8.64	31.86	32.76
	GRPO-prefix	91.78	66.40	19.12	64.14	54.71	60.90	40.20	9.19	**33.36**	**33.60**
	GRPO+IMAX	**94.01**	**67.20**	20.96	**67.06**	51.57	**64.18**	41.00	10.02	32.42	30.59
	DAPO-prefix	90.07	63.60	22.43	65.47	53.79	52.12	34.60	9.56	31.86	31.05
	DAPO+IMAX	92.72	67.00	**23.16**	63.55	51.76	60.20	**41.10**	**11.40**	32.24	30.45

two prefixes. At evaluation time, all prefixes are removed, so any improvement reflects the resulting policy rather than additional test-time conditioning.

5.2 Main Results

Inference-time diversity and quality. Table 1 shows that optimizing only the prefix pool improves inference-time performance over the base model on most benchmarks. These gains hold across all three backbones, which remain frozen during prefix-pool training. On MATH-500, GRPO+IMAX improves Pass@4 by 10.40, 17.20, and 8.60 percentage points on Qwen2.5-1.5B-Instruct, Qwen3-4B-Base, and Qwen3-8B-Base, respectively. The corresponding Avg@4 improvements are 9.45, 20.45, and 7.35 percentage points. Overall, IMAX delivers substantial inference-time gains over the frozen base models and improves both Pass@4 and Avg@4 over GRPO-prefix across most backbone–benchmark pairs. Because all prefix pools are trained only on MATH, the gains on GSM8K, Minerva Math, and AMC also demonstrate cross-dataset transfer.

Downstream RLVR exploration. We next test whether the learned prefixes remain useful beyond inference. For each variant in Table 2, we freeze its prefix pool and use the prefixes only to diversify rollout generation during a subsequent full-policy GRPO stage. The resulting policy is evaluated without any prefix, separating the effect of improved training exploration from test-time conditioning. IMAX-conditioned training outperforms standard full-policy GRPO on every reported metric. It also gives the strongest Pass@K among all rollout conditions on five benchmarks and the strongest Avg@K on four of them, with the largest improvements appearing on AMC23, AIME24, and AIME25. Because prefixes are absent at evaluation time, these gains show that the learned prefixes indeed provide more informative training rollouts rather than merely acting as test-time prompts.

Table 2: Downstream full-policy RLVR on Qwen3-8B. Prefixes are frozen and used only during rollout generation. Evaluation is performed without prefixes. We report percentages (%), with $K = 4$ for GSM8K, MATH-500, and AMC23, and $K = 16$ for AIME24 and AIME25.

	Pass@K					Avg@K				
Rollout condition	**GSM8K**	**MATH-500**	**AMC23**	**AIME24**	**AIME25**	**GSM8K**	**MATH-500**	**AMC23**	**AIME24**	**AIME25**
Full-policy GRPO	95.68	76.40	90.00	56.67	46.67	92.91	69.50	64.38	28.54	25.83
+ Random prefixes	95.91	76.60	80.00	43.33	43.33	92.97	69.55	50.00	31.46	21.25
+ Text-initialized prefixes	96.06	76.80	85.00	56.67	43.33	**94.03**	71.20	71.88	30.83	25.00
+ GRPO-trained prefixes	96.21	77.00	80.00	63.33	40.00	93.52	70.40	68.75	35.21	20.42
+ IMAX-trained prefixes	**96.82**	**77.60**	**92.50**	**73.33**	**63.33**	93.35	**71.75**	**75.63**	**41.04**	**29.17**

Table 3: Inference-time prefix-pool ablations on Qwen3-8B. (a) Initialization sensitivity for prefix-only IMAX (Pass@4/Avg@4). (b) IMAX prefix scaling on AMC23. All values are percentages.

(a) Sensitivity to prefix initialization				**(b) Prefix scaling on AMC23**			
Method	**GSM8K**	**MATH-500**	**AMC23**	**Method**	**Prefixes**	**Pass@64**	**Avg@128**
Full-policy GRPO	95.68 / 92.91	76.40 / 69.50	90.00 / 64.38	IMAX (text init.)	2	98.66	52.54
IMAX (random init.)	94.47 / 84.67	76.00 / 65.45	70.00 / 46.88		4	99.44	54.23
IMAX (text init.)	**96.29 / 91.21**	**76.40 / 66.80**	**72.50 / 54.38**		8	100.00	52.80

5.3 Ablation Studies

Sensitivity to initialization. Table 3 (a) compares IMAX with text and random initialization under the same prefix-only training protocol. Text initialization improves all reported prefix-only metrics. Random initialization remains close in Pass@4 on GSM8K and MATH-500 but incurs larger Avg@4 drops, especially on GSM8K and AMC23. Text initialization consistently improves both Pass@4 and Avg@4 over random initialization across all three benchmarks. This shows that initialization meaningfully shapes the behaviors elicited from the frozen model, with semantically grounded prefixes providing a stronger starting point than random vectors.

Inference-time prefix scaling. Table 3 (b) compares 2, 4, and 8 prefixes on AMC23 with eight training rollouts per prefix. The optimization steps are 300, 150, and 75, respectively, to control the total rollout budgets to be the same as well as keeping the group size for each prefix the same. Pass@64 rises from 98.66 to 99.44 and 100.00, while Avg@128 peaks at 54.23 with four prefixes. Thus, larger prefix pools improve high-budget solution coverage by distributing rollouts across a broader set of learned reasoning modes. The marginal Pass@64 gain from four to eight prefixes, together with the drop in Avg@128, suggests diminishing returns as the fixed rollout budget is distributed across more prefix-conditioned modes.

5.4 Structured Reasoning Analysis

We first list the two initialized prefixes used in this analysis, so that the subsequent comparisons of reasoning components can be interpreted with respect to their intended semantic roles.

> **Prefix 1.** Introduce variables explicitly, write down the governing equations or identities, and solve through exact symbolic manipulation. Prefer precise derivation over heuristic guessing.
>
> **Prefix 2.** Partition the problem into exhaustive, mutually exclusive cases. Solve each case carefully, rule out impossible ones, and combine the valid conclusions.

IMAX induces structured reasoning diversity. We next examine whether IMAX induces distinct reasoning behaviors within the prefix group. Figure 2 shows that IMAX yields more separated correct-answer embedding clusters across prefixes, indicating stronger intra-group diversity among successful trajectories. This diversity is also semantically aligned with the prefix designs: in Figure 3a, Prefix 1 induces more equation, algebra, and arithmetic operations, while Prefix 2 induces more subgoals, cases, backtracking, and verification. The contrast is strongest for subgoals and verification, where IMAX recovers the intended prefix-specific pattern more clearly than competing methods.

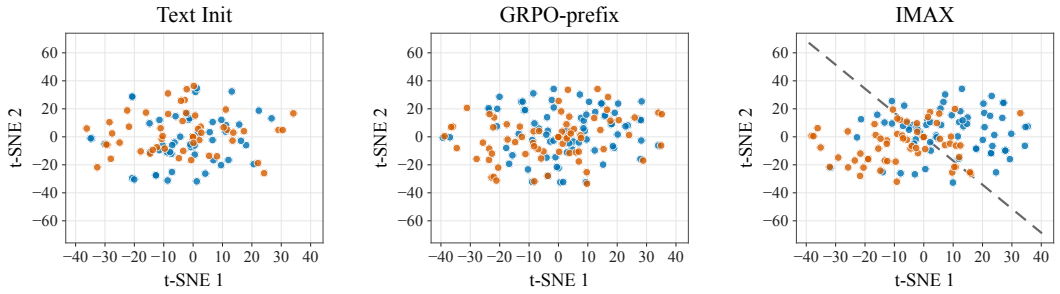
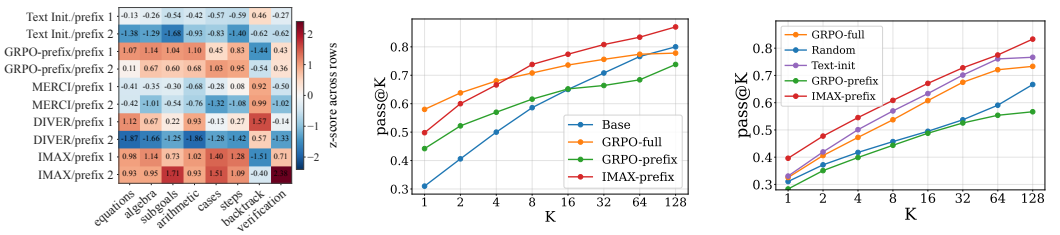


Figure 2: t-SNE visualization of correct-response embeddings for the two trained prefixes with Qwen3-4B-Base. Blue and orange points denote Prefix 1 and Prefix 2, respectively. The complete five-method comparison is provided in Figure 7 in Appendix C.



(a) Trajectory visualization. (b) Inference-time Pass@K (c) Downstream RLVR Pass@K.

Figure 3: Analysis of IMAX trajectories. (a) Trajectory visualization between two learned prefixes. (b) Inference-time Pass@K on MATH-500 with Qwen3-4B-Base. (c) AIME24 Pass@K after downstream full-policy RLVR on Qwen3-8B. Additional analyses are provided in Appendix C.

IMAX enhances effective exploration. At inference time, Figure 3b shows that GRPO-full is strongest at small K on MATH-500, whereas IMAX surpasses it when $K > 4$ and remains above the base model, indicating less redundant coverage of correct solutions. The benefit transfers to downstream RLVR: IMAX-conditioned training achieves the highest AIME24 Pass@K at every budget, improving Pass@128 by 10.00 points over full-policy GRPO, while Table 2 reports a 12.50-point Avg@16 gain. Together, these results show that the learned prefixes improve inference-time coverage and downstream exploration without sacrificing rollout quality.

Runtime efficiency analysis. We report the runtime for each method in Figure 10 at Appendix C, due to limited space in the main context. The additional runtime of IMAX over standard prefix-tuned RLVR baselines mainly comes from the extra forward pass used to obtain the prompt–response representation for training the posterior model q_ϕ. Even with this overhead, IMAX remains substantially cheaper than full-parameter GRPO (measured in GPU-hours). IMAX requires about $3.0\times$–$6.0\times$ less training cost across model scales. This advantage becomes more important as the backbone grows, because prefix-tuned methods only incur increased rollout cost from the base model, whereas full-parameter tuning also scales the backward and optimizer costs over all model parameters.

6 Conclusion

In this work, we study effective exploration in RLVR for LLM reasoning from the perspective of promoting diverse reasoning paths. We propose Information-Maximizing Augmented eXploration (IMAX), a lightweight post-training framework that optimizes a pool of soft prompts to reshape the model's conditional generation distribution. To prevent the learned prefixes from collapsing into redundant controls, we introduce a response-level InfoMax reward, and optimize the posterior estimator as well as the prefix pool in an alternating procedure. Experiments on mathematical reasoning benchmarks show that IMAX improves both reasoning quality and prefix-conditioned diversity over standard RLVR baselines. Learned prefixes further improve both inference-time reasoning and downstream RLVR exploration, yielding stronger prefix-free policies.

AI USE STATEMENT

In this work, we used generative AI tools for none of the tasks with required disclosure. We have not used generative AI tools for any other tasks with required disclosure, and the rest of the required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools to edit a research paper to improve readability and create or modify scientific figures or images. We have reviewed all AI-assisted work. All AI-assisted edits and artifacts were manually reviewed and verified by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The experimental setup and evaluation protocol are described in Section 5.1, and the complete training procedure is summarized in Algorithm 1. Additional implementation details and the default prefix pool are provided in Appendix D. Dataset and code licenses are listed in Appendix E. Following the official ICLR guidelines, we will supply an anonymous repository containing our code and experiment scripts via a comment directed to the reviewers and area chairs. All materials will be open-sourced upon publication.

REFERENCES

Bizhe Bai, Xinyue Wang, Peng Ye, and Tao Chen. Learning to explore with parameter-space noise: A deep dive into parameter-space noise for reinforcement learning with verifiable rewards. *arXiv preprint arXiv:2602.02555*, 2026.

David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems*, 29, 2016.

Zhipeng Chen, Yingqian Min, Beichen Zhang, Jie Chen, Jinhao Jiang, Daixuan Cheng, Wayne Xin Zhao, Zheng Liu, Xu Miao, Yang Lu, et al. An empirical study on eliciting and improving r1-like reasoning models. *arXiv preprint arXiv:2503.04548*, 2025a.

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*, 2025b.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

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.

Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric P. Xing, and Zhiting Hu. RLPrompt: Optimizing discrete text prompts with reinforcement learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 3369–3391, 2022.

Jingchu Gai, Guanning Zeng, Huaqing Zhang, and Aditi Raghunathan. Differential smoothing mitigates sharpening and improves llm reasoning. *arXiv preprint arXiv:2511.19942*, 2025.

Jingtong Gao, Ling Pan, Yejing Wang, Rui Zhong, Chi Lu, Maolin Wang, Qingpeng Cai, Peng Jiang, and Xiangyu Zhao. Navigate the unknown: Enhancing llm reasoning with intrinsic motivation guided exploration. *arXiv preprint arXiv:2505.17621*, 2025.

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

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

Zican Hu, Shilin Zhang, Yafu Li, Jianhao Yan, Xuyang Hu, Leyang Cui, Xiaoye Qu, Chunlin Chen, Yu Cheng, and Zhi Wang. Diversity-incentivized exploration for versatile reasoning. *arXiv preprint arXiv:2509.26209*, 2025.

Guanhua Huang, Tingqiang Xu, Mingze Wang, Qi Yi, Xue Gong, Siheng Li, Ruibin Xiong, Kejiao Li, Yuhao Jiang, and Bo Zhou. Low-probability tokens sustain exploration in reinforcement learning with verifiable reward. *arXiv preprint arXiv:2510.03222*, 2025.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Aayush Karan and Yilun Du. Reasoning with sampling: Your base model is smarter than you think. *arXiv preprint arXiv:2510.14901*, 2025.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, 2021.

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.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597, 2021.

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 *The twelfth international conference on learning representations*, 2023.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-Tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pp. 61–68, 2022.

Aleksandar Petrov, Philip H. S. Torr, and Adel Bibi. When do prompting and prefix-tuning work? a theory of capabilities and limitations. In *International Conference on Learning Representations*, 2024.

Guanghui Qin and Jason Eisner. Learning how to ask: Querying LMs with mixtures of soft prompts. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5203–5212, 2021.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

Devan Shah, Owen Yang, Daniel Yang, Chongyi Zheng, and Benjamin Eysenbach. Upskill: Mutual information skill learning for structured response diversity in llms. *arXiv preprint arXiv:2602.22296*, 2026.

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.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. AutoPrompt: Eliciting knowledge from language models with automatically generated prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 4222–4235, 2020.

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.

James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11909–11919, 2023.

Hongze Tan, Zihan Wang, Jianfei Pan, Jinghao Lin, Hao Wang, Yifan Wu, Tao Chen, Zhihang Zheng, Zhihao Tang, and Haihua Yang. Gtpo and grpo-s: Token and sequence-level reward shaping with policy entropy. *arXiv preprint arXiv:2508.04349*, 2025.

Tu Vu, Brian Lester, Noah Constant, Rami Al-Rfou, and Daniel Cer. SPoT: Better frozen model adaptation through soft prompt transfer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pp. 5039–5059, 2022.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*, 2025.

Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *European conference on computer vision*, pp. 631–648. Springer, 2022a.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 139–149, 2022b.

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*, 2025.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024a.

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*, 2024b.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

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. *arXiv preprint arXiv:2503.14476*, 2025.

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

Xuan Zhang, Ruixiao Li, Zhijian Zhou, Long Li, Yulei Qin, Ke Li, Xing Sun, Xiaoyu Tan, Chao Qu, and Yuan Qi. Count counts: Motivating exploration in llm reasoning with count-based intrinsic rewards. *arXiv preprint arXiv:2510.16614*, 2025a.

Xuan Zhang, Ruixiao Li, Zhijian Zhou, Long Li, Yulei Qin, Ke Li, Xing Sun, Xiaoyu Tan, Chao Qu, and Yuan Qi. Count counts: Motivating exploration in llm reasoning with count-based intrinsic rewards. *arXiv preprint arXiv:2510.16614*, 2025b.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025a.

Tianyu Zheng, Tianshun Xing, Qingshui Gu, Taoran Liang, Xingwei Qu, Xin Zhou, Yizhi Li, Zhoufutu Wen, Chenghua Lin, Wenhao Huang, Qian Liu, Ge Zhang, and Zejun Ma. First return, entropy-eliciting explore. *arXiv preprint arXiv:2507.07017*, 2025b.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

Xinyu Zhu, Mengzhou Xia, Zhepei Wei, Wei-Lin Chen, Danqi Chen, and Yu Meng. The surprising effectiveness of negative reinforcement in llm reasoning. *arXiv preprint arXiv:2506.01347*, 2025.

Appendix

A Complete Related Works

We first review RLVR algorithms for LLM reasoning and then discuss the exploration challenge in RLVR. We next introduce soft prompt tuning, which forms the basis of our setting. Finally, we compare IMAX with concurrent mutual-information-based exploration work.

Reinforcement learning from verifiable rewards. As large language models (LLMs) show strong capabilities across tasks, improving their reasoning ability has become critical. Among existing approaches, reinforcement learning from verifiable rewards (RLVR) has attracted growing attention for enhancing LLM reasoning with verifiable feedback. Early reinforcement learning demonstrated its potential for reasoning tasks with the release of the OpenAI o1 model Jaech et al. (2024). Subsequently, the first open-source reasoning model, DeepSeek-R1 (Guo et al., 2025), was released, achieving comparable performance. Earlier, DeepSeekMath (Shao et al., 2024) introduced group relative policy optimization (GRPO), which has since become a widely used RLVR algorithm for LLM post-training. These developments have sparked a wave of research on RLVR for LLM reasoning (Yu et al., 2025; Zheng et al., 2025a; Tan et al., 2025). Despite the rapid progress of RLVR, this paradigm still faces challenges. Several studies have reported an "entropy collapse" phenomenon of RLVR (Cui et al., 2025; Wang et al., 2025). Moreover, debates on whether RLVR genuinely enhances reasoning or merely increases confidence in the original distribution support the concern that RLVR may drive models toward determinism rather than improved exploration (Yue et al., 2025; Zhu et al., 2025).

Exploration in RLVR. Exploration is a central difficulty in RLVR because verifiable rewards are sparse and are typically observed only after complete reasoning trajectories are sampled. Recent analyses show that RLVR can rapidly concentrate probability mass and that effective learning often depends on preserving informative high-entropy or low-probability decision points, rather than uniformly increasing randomness (Cui et al., 2025; Wang et al., 2025; Huang et al., 2025; Gai et al., 2025; Chen et al., 2025a). One line of work addresses this issue by modifying the optimization objective, such as DAPO (Yu et al., 2025) and Pass@K-oriented training that rebalance exploitation and exploration (Chen et al., 2025b). Another line introduces explicit exploration bonuses or diversity incentives through intrinsic rewards (Gao et al., 2025; Zhang et al., 2025a; Hu et al., 2025). Other methods intervene in the rollout process, for example by targeting entropy-eliciting tokens (Zheng et al., 2025b) or injecting parameter-space noise (Bai et al., 2026). Together, these approaches improve exploration by modifying the objective, reward, or rollout dynamics of the policy. Our work takes a complementary route: it learns multiple input-level conditioning priors, each of which reshapes the rollout distribution before generation.

Soft prompting. Prompt-based adaptation spans several mechanisms: hard prompt search optimizes discrete tokens in the vocabulary (Shin et al., 2020; Deng et al., 2022), soft prompt tuning learns continuous virtual tokens prepended to the input embedding sequence (Lester et al., 2021; Liu et al., 2022), and prefix-tuning injects trainable prefix key-value states into each Transformer layer for generation (Li & Liang, 2021). These methods support the view that prompts act as lightweight control variables for frozen models: mixtures of soft prompts can query different latent behaviors from the same LM (Qin & Eisner, 2021), source prompts can transfer across tasks (Vu et al., 2022), and theoretical analyses suggest that prompting and prefix-tuning often work by eliciting capabilities already present in the pretrained model rather than rewriting the model's internal computation (Petrov et al., 2024). Prompt-pool methods further show that collections of prompts can serve as compact task memories or input-conditioned control modules (Wang et al., 2022b;a; Smith et al., 2023). However, existing RL-based prompt optimization mainly targets discrete prompt search (Deng et al., 2022), and diversity-aware prompt pools have been studied primarily for task selection or continual learning rather than for RLVR reasoning rollouts. Our work instead optimizes a pool of soft prompts with verifiable rewards and an InfoMax reward so that different prompts induce distinct, high-quality reasoning trajectories. Throughout this paper, we call each learned input-level soft prompt a *prefix*; this usage is distinct from layer-wise prefix-tuning (Li & Liang, 2021).

Comparison with concurrent MI-based RLVR exploration. Concurrent work UpSkill (Shah et al., 2026) also uses a mutual-information objective to induce structured response diversity during RLVR. The methods differ in where this control is parameterized: UpSkill learns latent skills through policy optimization, whereas IMAX freezes the backbone and optimizes a pool of input-level soft prompts initialized from textual strategy descriptions. The learned prompts therefore remain explicit and reusable conditioning variables: they can broaden inference-time solution coverage and condition rollouts for subsequent full-policy RLVR, while the resulting policy can still be evaluated without them. This design reflects IMAX's focus on constructing multiple controllable reasoning priors, with the response-level InfoMax reward discouraging those priors from collapsing to redundant rollout behaviors.

B Discussion on the InfoMax Reward

In the InfoMax reward design, we choose the reverse form of the mutual information to derive the variational lower bound. For the forward form, consider the following decomposition of the mutual information objective:

$$\begin{aligned} I(Z;Y \mid X) &= H(Y \mid X) - H(Y \mid X, Z) \\ &= \mathbb{E}_{p(x)}\big[\mathbb{E}_{p(y|x)}\big[-\log p(y \mid x)\big]\big] - \mathbb{E}_{p(x)}\big[\mathbb{E}_{p(z,y|x)}\big[-\log p(y \mid x, z)\big]\big] \\ &= \mathbb{E}_{p(x)}\big[\mathbb{E}_{p(z,y|x)}\big[\log p(y \mid x, z)\big] - \mathbb{E}_{p(y|x)}\big[\log p(y \mid x)\big]\big], \\ \text{where} \quad & p(z, y \mid x) = p(z \mid x)\, p(y \mid x, z), p(y \mid x) = \mathbb{E}_{p(z|x)}[p(y \mid x, z)]. \end{aligned}$$

In LLMs, $p(y \mid x, z)$ is the conditional probability assigned to response y given question x and latent prefix z. However, the second term is an intractable marginalization term over the latent prefix z,

$$\mathbb{E}_{p(y|x)}\big[\log p(y \mid x)\big] = \mathbb{E}_{p(y|x)}\left[\log \sum_{z=1}^{C} p(z \mid x) p(y \mid x, z)\right].$$

A biased contrastive target. It is worth noting that under the latent-variable formulation, this term does not correspond to a single LLM forward pass on (x, y). Rather, it implicitly involves marginalizing over the latent prefix, evaluating how well the output can be explained across different prefix-conditioned models induced by the prior. This is because the model defines $p(y \mid x)$ only after marginalizing over the latent prefix. Concretely, the LLM induces a family of prefix-conditioned distributions, and the likelihood of y is obtained by aggregating their contributions under the prior rather than evaluating a single prefix instance. As a result, this term cannot be computed by a single forward pass on (x, y) with a fixed or absent latent variable. If this term is instead approximated by a single forward pass with a fixed or ignored latent prefix, the resulting objective no longer corresponds to the true mutual information. In this case, it effectively becomes a *contrastive target*: it measures the gap between the likelihood under a *single* prefix-conditioned model $p(y \mid x, z)$ (or an implicitly fixed z) and the likelihood under the prefix-marginalized model $p(y \mid x)$. Intuitively, this contrasts "with z" versus "without z", rather than capturing the dependence between z and y averaged over the latent space. Optimizing such an objective tends to encourage the policy to exploit a particular z (or ignore z altogether), leading to biased gradients, latent-space collapse, and reduced diversity of prefix-coupled reasoning trajectories.

Intractable marginalization. Approximating this term requires sampling multiple latent prefixes per question to account for marginalization, which substantially increases the sampling cost. This burden is further amplified in group-relative RLVR algorithms that already rely on multiple rollouts per prompt. Consequently, we do not adopt this form of mutual information when deriving our InfoMax reward.

C Additional Experiment Results

This appendix provides the experimental evidence omitted from the main text for space. We first report the training-free prefix control and complete benchmark tables, then provide downstream RLVR,

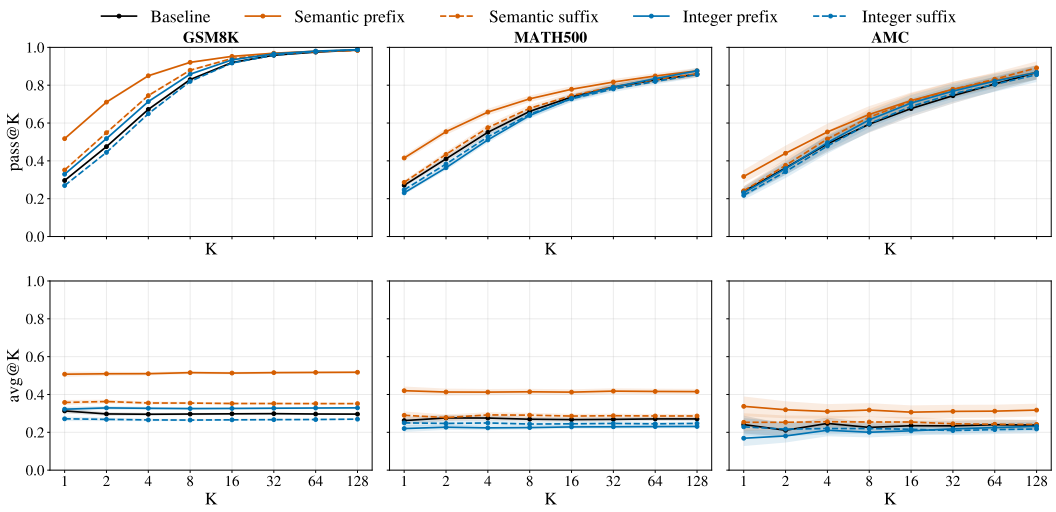


Figure 4: Inference-time performance of training-free prompting. We compare the base model with semantic and integer prompts inserted either as prefixes or suffixes across GSM8K, MATH-500, and AMC on Qwen2.5-MATH-1.5B. The top row reports Pass@K, while the bottom row reports Avg@K.

initialization, and prefix-scaling results, followed by additional visual analyses of prefix-conditioned behavior and runtime cost.

C.1 Training-Free Prefix Controls

We first test whether changing the generation condition can shift the inference behavior of a frozen Qwen2.5-MATH-1.5B model (Yang et al., 2024b). Figure 4 compares semantic and integer prompts inserted as prefixes or suffixes. Semantic prefixes improve Pass@K most clearly at small to moderate sampling budgets and also improve Avg@K across the three benchmarks. This control motivates prefix-level conditioning but does not involve IMAX training; the matched comparisons in the main experiments isolate the additional contribution of RL and the InfoMax reward.

C.2 Main Results

Tables 4, 5, and 6 report the complete Pass@4 and Avg@4 results of Qwen2.5-1.5B-Instruct, Qwen3-4B-Base, and Qwen3-8B-Base, respectively. All values are percentages, and standard errors are reported for both metrics except for the corrected GRPO-full AMC point estimates on Qwen2.5-1.5B-Instruct and Qwen3-4B-Base. These tables expand Table 1 by including the full benchmark set, including GPQA, AIME24, and AIME25.

For M evaluation examples, we compute the standard error over per-example scores s_i:

$$\bar{s} = \frac{1}{M}\sum_{i=1}^{M} s_i, \qquad \mathrm{stderr}(s) = \sqrt{\frac{1}{M(M-1)}\sum_{i=1}^{M}(s_i - \bar{s})^2}. \tag{9}$$

For Pass@K, $s_i = \mathbf{1}[\exists j \le K : c_{ij} = 1]$; for Avg@$K$, $s_i = \frac{1}{K_i}\sum_{j=1}^{K_i} c_{ij}$, where $c_{ij} \in \{0,1\}$ indicates whether rollout j for example i is correct and $K_i = \min(K, \text{number of rollouts for example } i)$. The reported Pass@$K$ and Avg@K values are the means $\frac{1}{M}\sum_i s_i$ under the corresponding per-example definition.

C.3 Additional Downstream Results

Table 7 expands Table 2 by reporting Pass@K over sampling budgets from 1 to 128 on AMC23 and AIME24. Each value is estimated from 128 samples per problem, and the trained policies are

Table 4: Full results for Qwen2.5-1.5B-Instruct.

Method	GSM8K	MATH	Minerva	GPQA	AMC	AIME24	AIME25	IFEval
Pass@4								
Base	78.92±1.33	49.60±2.24	13.97±2.11	62.63±3.45	68.22±1.35	13.33±6.22	0.00±0.00	52.31±2.15
GRPO-full	84.84±0.99	60.20±2.19	18.01±2.33	60.10±3.49	68.06	13.33±6.31	6.67±4.63	50.28±0.00
Text Init.	80.29±1.30	50.20±2.24	16.18±2.24	64.65±3.41	68.31±1.35	6.67±4.63	6.67±4.63	54.34±2.14
DIVER-prefix	80.29±1.10	56.20±2.22	16.54±2.26	65.66±3.38	67.72±1.35	6.67±4.63	3.33±3.33	54.90±2.14
MERCI-prefix	82.11±1.06	59.00±2.20	16.18±2.24	57.58±3.52	68.31±1.34	3.33±3.33	0.00±0.00	54.53±2.14
GRPO-prefix	85.75±1.17	49.40±2.24	15.44±2.20	60.61±3.48	69.22±1.34	0.00±0.00	6.67±4.63	53.60±2.15
GRPO-prefix+IMAX	85.75±1.17	60.00±2.19	18.38±2.35	62.12±3.46	68.06±1.35	0.00±0.00	3.33±3.33	53.05±2.15
DAPO-prefix	86.28±0.95	56.40±2.22	17.28±2.30	65.15±3.39	68.97±1.34	10.00±5.57	0.00±0.00	51.94±0.00
DAPO-prefix+IMAX	84.99±0.98	59.40±2.20	15.81±2.22	60.10±3.49	68.39±1.34	6.67±4.63	6.67±4.63	53.79±0.00
Avg@4								
Method	**GSM8K**	**MATH**	**Minerva**	**GPQA**	**AMC**	**AIME24**	**AIME25**	**IFEval**
Base	53.43±1.23	32.85±1.76	6.99±1.22	26.77±1.93	35.28±0.96	3.33±1.56	0.00±0.00	37.75±1.81
GRPO-full	74.09±1.05	47.55±2.00	10.94±1.61	28.41±2.12	34.45	5.83±3.10	2.50±1.84	35.26±0.00
Text Init.	56.92±1.23	34.35±1.81	8.82±1.40	28.03±1.92	34.70±0.94	1.67±1.16	1.67±1.16	38.08±1.77
DIVER-prefix	59.76±1.08	37.05±1.74	7.72±1.24	27.53±1.87	33.28±0.91	2.50±1.84	0.83±0.83	38.03±1.76
MERCI-prefix	63.34±1.07	40.90±1.83	8.64±1.35	23.61±1.83	35.05±0.95	1.67±1.67	0.00±0.00	37.43±1.75
GRPO-prefix	66.89±1.22	34.80±1.84	8.27±1.35	27.15±1.94	35.38±0.95	0.00±0.00	1.67±1.16	37.11±1.75
GRPO-prefix+IMAX	67.40±1.22	42.30±1.86	9.47±1.43	26.77±1.97	34.45±0.95	0.00±0.00	0.83±0.83	36.78±1.75
DAPO-prefix	69.50±1.02	41.60±1.91	9.19±1.41	27.40±1.83	35.53±0.95	4.17±2.42	0.00±0.00	36.74±0.00
DAPO-prefix+IMAX	67.63±1.03	42.50±1.88	9.83±1.57	25.38±1.91	34.99±0.95	1.67±1.16	1.67±1.16	38.12±0.00

Table 5: Full results for Qwen3-4B-Base.

Method	GSM8K	MATH	Minerva	GPQA	AMC	AIME24	AIME25	IFEval
Pass@4								
Base	90.60±0.80	52.00±2.24	22.43±2.53	68.69±3.30	54.13±1.44	10.00±5.57	3.33±3.33	48.80±2.15
GRPO-full	91.81±0.76	69.20±2.07	25.37±2.64	52.53±3.56	71.14	10.00±5.57	13.33±6.31	43.25±0.00
Text Init.	88.55±0.88	51.20±2.24	24.26±2.60	58.08±3.52	67.39±1.35	16.67±6.92	3.33±3.33	50.65±2.15
DIVER-prefix	91.43±0.77	57.80±2.21	23.53±2.58	59.60±3.50	66.97±1.36	6.67±4.63	10.00±5.57	52.31±2.15
MERCI-prefix	90.83±0.80	60.80±2.19	20.96±2.47	52.53±3.56	64.80±1.38	0.00±0.00	3.33±3.33	51.20±0.00
GRPO-prefix	93.48±0.68	57.60±2.21	21.69±2.50	58.08±3.52	70.64±1.32	13.33±6.31	13.33±6.31	51.39±2.15
GRPO-prefix+IMAX	94.47±0.63	69.20±2.07	23.90±2.59	61.11±3.47	71.06±1.31	13.33±6.31	6.67±4.63	51.57±2.15
DAPO-prefix	93.18±0.69	56.60±2.22	25.74±2.66	55.56±3.54	69.06±1.34	20.00±7.43	0.00±0.00	56.19±2.14
DAPO-prefix+IMAX	93.78±0.67	65.80±2.12	23.90±2.59	58.08±3.52	64.97±1.38	13.33±6.31	10.00±5.57	55.08±0.00
Avg@4								
Method	**GSM8K**	**MATH**	**Minerva**	**GPQA**	**AMC**	**AIME24**	**AIME25**	**IFEval**
Base	59.70±0.87	30.00±1.55	10.57±1.39	24.49±1.53	22.83±0.76	3.33±1.98	1.67±1.67	26.94±1.46
GRPO-full	86.64±0.84	59.80±2.01	20.13±2.28	25.38±2.21	44.02	7.50±4.67	3.33±1.58	29.25±0.00
Text Init.	53.49±0.84	27.50±1.47	10.02±1.25	28.54±2.22	35.22±0.96	6.67±2.92	0.83±0.83	28.97±1.51
DIVER-prefix	66.49±0.87	31.75±1.49	9.93±1.27	27.02±2.05	35.38±0.96	2.50±1.84	2.50±1.39	30.41±1.53
MERCI-prefix	62.51±0.86	30.75±1.41	9.56±1.26	23.11±1.94	34.92±0.98	0.00±0.00	0.83±0.83	31.47±0.00
GRPO-prefix	71.97±0.81	43.45±1.89	13.33±1.74	26.39±2.07	39.64±1.00	5.83±3.10	4.17±2.10	31.93±1.61
GRPO-prefix+IMAX	76.14±0.81	50.45±1.81	11.86±1.46	25.25±1.89	38.07±0.96	5.00±2.79	2.50±1.84	30.64±1.58
DAPO-prefix	72.02±0.84	41.65±1.87	14.52±1.73	24.37±1.98	38.32±1.00	5.83±2.30	0.00±0.00	35.95±1.67
DAPO-prefix+IMAX	72.93±0.82	42.70±1.69	10.85±1.37	23.99±1.93	35.78±1.00	5.83±2.86	2.50±1.39	33.13±0.00

evaluated without prefixes. IMAX-conditioned full-policy training gives the strongest Pass@K across all reported budgets on AIME24 and across $K \leq 64$ on AMC23; the prefix-conditioned variants converge at 97.50% Pass@128 on AMC23.

Fixed IMAX prefixes mitigate entropy collapse during downstream RLVR. Figure 5 tracks policy entropy throughout the stage-2 full-policy GRPO training described in Section 5.2. The entropy of standard full-policy GRPO falls to a low level late in training, whereas rollout conditioning with a fixed IMAX prefix pool maintains substantially higher entropy throughout the same stage and remains highest in the late-training region. The comparison with random, text-initialized, and GRPO-trained prefix pools further suggests that the effect is not explained by prefix conditioning alone: among the compared conditions, the fixed IMAX prefixes provide the strongest mitigation of entropy collapse while the full policy is optimized.

C.4 Additional Ablations

Table 8 reports the complete one-, two-, and four-prefix results with standard errors for both Pass@4 and Avg@4. All values are percentages. The per-prefix group size is fixed across these runs, while the number of optimization steps is set to 600, 300, and 150, respectively, matching the total number of sampled training trajectories.

Table 6: Full results for Qwen3-8B-Base.

Pass@4								
Method	**GSM8K**	**MATH**	**Minerva**	**GPQA**	**AMC**	**AIME24**	**AIME25**	**IFEval**
Base	88.10±0.89	58.60±2.20	18.01±2.33	29.29±3.24	41.12±1.42	13.33±6.31	10.00±5.57	48.06±2.15
GRPO-full	91.58±0.76	74.20±1.96	29.41±2.77	47.98±3.56	74.23±1.26	26.67±8.21	33.33±8.75	47.87±0.00
Text Init.	90.30±0.82	59.80±2.19	20.96±2.47	55.56±3.54	62.39±1.40	20.00±7.43	6.67±4.63	52.50±2.15
DIVER-prefix	90.75±0.80	63.00±2.16	21.69±2.50	47.47±3.56	62.89±1.40	10.00±5.57	6.67±4.63	55.27±2.14
MERCI-prefix	93.18±0.69	66.40±2.11	18.38±2.35	55.56±3.54	66.47±1.36	10.00±5.57	6.67±4.63	54.71±2.14
GRPO-prefix	91.78±0.87	66.40±2.11	19.12±2.39	50.00±3.56	64.14±1.39	16.67±6.92	3.33±3.33	54.71±2.14
GRPO-prefix+IMAX	94.01±0.65	67.20±2.10	20.96±2.47	55.56±3.54	67.06±1.36	10.00±5.57	10.00±5.57	51.57±2.15
DAPO-prefix	90.07±0.82	63.60±2.15	22.43±2.53	48.48±3.56	65.47±1.37	10.00±5.57	6.67±4.63	53.79±2.15
DAPO-prefix+IMAX	92.72±0.72	67.00±2.10	23.16±2.56	54.04±3.55	63.55±1.39	13.33±6.31	16.67±6.92	51.76±2.15
Avg@4								
Method	**GSM8K**	**MATH**	**Minerva**	**GPQA**	**AMC**	**AIME24**	**AIME25**	**IFEval**
Base	55.44±0.86	33.65±1.60	8.73±1.31	8.59±1.03	18.58±0.79	4.17±2.10	3.33±1.98	27.26±1.50
GRPO-full	76.95±0.89	65.30±1.93	22.33±2.31	19.57±1.78	47.46±1.09	16.67±5.91	16.67±5.41	31.24±0.00
Text Init.	56.16±0.82	33.60±1.52	8.36±1.11	22.73±1.81	31.15±0.91	5.83±2.30	2.50±1.84	30.87±1.55
DIVER-prefix	57.15±0.80	36.65±1.60	9.56±1.26	17.05±1.50	29.69±0.86	3.33±1.98	1.67±1.16	32.76±1.58
MERCI-prefix	59.59±0.79	37.95±1.53	8.64±1.24	20.83±1.68	31.86±0.87	3.33±1.98	2.50±1.84	32.76±1.59
GRPO-prefix	60.90±0.82	40.20±1.61	9.19±1.31	17.93±1.51	33.36±0.94	5.83±2.59	0.83±0.83	33.60±1.62
GRPO-prefix+IMAX	64.18±0.79	41.00±1.64	10.02±1.35	22.73±1.84	32.42±0.87	3.33±1.98	3.33±1.98	30.59±1.54
DAPO-prefix	52.12±0.79	34.60±1.48	9.56±1.24	18.18±1.60	31.86±0.89	3.33±1.98	2.50±1.84	31.05±1.54
DAPO-prefix+IMAX	60.20±0.77	41.10±1.64	11.40±1.47	22.22±1.84	32.24±0.93	4.17±2.10	6.67±2.92	30.45±1.54

Table 7: Pass@K (%) on Qwen3-8B after downstream full-policy training. Prefixes are used only during training rollouts and removed at evaluation.

Dataset	**Rollout condition**	**@1**	**@2**	**@4**	**@8**	**@16**	**@32**	**@64**	**@128**
AMC23	Full-policy GRPO	63.75	77.28	86.07	90.60	92.70	93.86	94.69	95.00
	+ Random prefixes	66.78	75.70	82.28	87.03	90.16	92.59	94.83	**97.50**
	+ Text-initialized prefixes	69.84	77.63	83.54	87.92	91.36	94.15	96.43	**97.50**
	+ GRPO-trained prefixes	62.03	73.98	84.11	87.75	90.33	92.60	95.10	**97.50**
	+ IMAX-trained prefixes	**72.36**	**80.56**	**87.15**	**91.87**	**94.65**	**96.01**	**96.88**	**97.50**
AIME24	Full-policy GRPO	32.55	40.60	47.26	53.73	60.76	67.50	72.09	73.33
	+ Random prefixes	31.07	37.19	41.74	45.72	49.50	53.72	59.07	66.67
	+ Text-initialized prefixes	33.07	41.93	50.09	56.97	63.35	70.14	76.07	76.67
	+ GRPO-trained prefixes	28.33	35.07	39.88	44.36	48.79	52.56	55.38	56.67
	+ IMAX-trained prefixes	**39.61**	**47.74**	**54.53**	**60.85**	**67.09**	**72.80**	**77.50**	**83.33**

The posterior becomes increasingly accurate during training. As shown in Figure 6, both pre-update and post-update prefix-identification accuracy begin near chance and increase as training proceeds, reaching above 80% for much of the later stage. Post-update accuracy generally matches or exceeds pre-update accuracy, showing that the alternating posterior update makes immediate use of the current prompt–response pairs, while the sustained increase in pre-update accuracy indicates that prefix-identifying information persists across training steps. Together, these trends show that q_ϕ progressively learns to recover the generating prefix from the response rather than remaining near chance, as it would if the posterior could not extract persistent prefix-identifying information.

C.5 Further Analysis

We provide additional figures that support the qualitative conclusions in Section 5.4. Figure 7 shows response-embedding structure across three reasoning benchmarks, Figure 8 summarizes reasoning-component differences between prefixes, Figure 9 compares complementary correctness across prefixes, and Figure 10 reports runtime cost.

For the embedding figures, we report a normalized centroid-separation score between `pool_001` and `pool_002` in the plotted two-dimensional t-SNE space. Let $\mathcal{V}_1 = \{\mathbf{v}_i : \text{point } i \text{ uses } \texttt{pool_001}\}$ and $\mathcal{V}_2 = \{\mathbf{v}_i : \text{point } i \text{ uses } \texttt{pool_002}\}$, where $\mathbf{v}_i \in \mathbb{R}^2$ is the plotted coordinate. We compute

$$\mu_k = \frac{1}{|\mathcal{V}_k|} \sum_{\mathbf{v} \in \mathcal{V}_k} \mathbf{v}, \quad s_k = \frac{1}{|\mathcal{V}_k|} \sum_{\mathbf{v} \in \mathcal{V}_k} \|\mathbf{v} - \mu_k\|_2, \quad \text{sep} = \frac{\|\mu_1 - \mu_2\|_2}{\max\left((s_1 + s_2)/2, 10^{-12}\right)}. \tag{10}$$

Higher separation therefore means that the two prefix clusters are farther apart relative to their average within-prefix spread. The plotted points are correct-only responses and are example-centered before t-SNE projection.

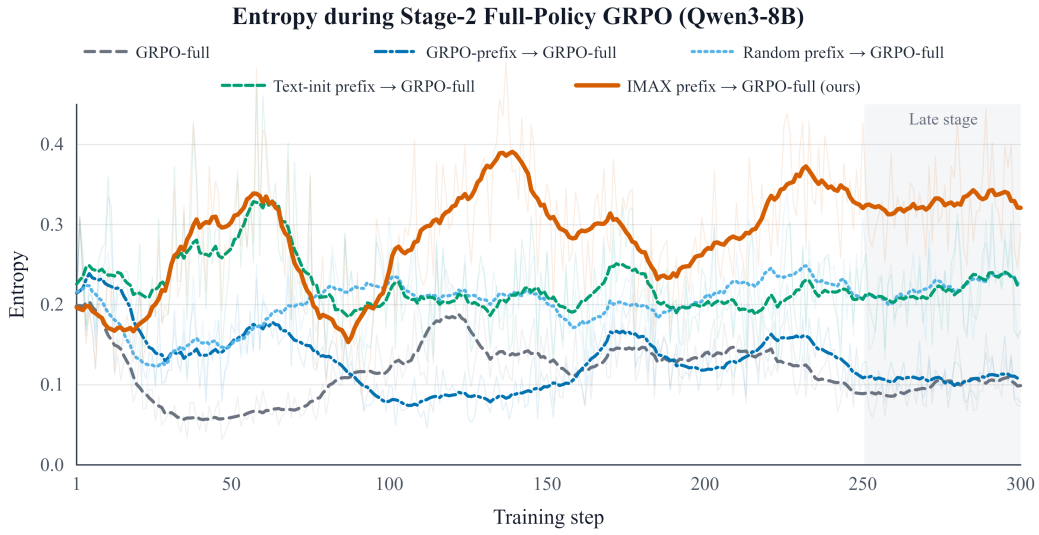


Figure 5: Policy entropy during downstream full-policy GRPO on Qwen3-8B. For the prefix-conditioned variants, the indicated prefix pool is fixed and used only to condition training rollouts. Faint curves show per-step values and opaque curves show the corresponding smoothed trends.

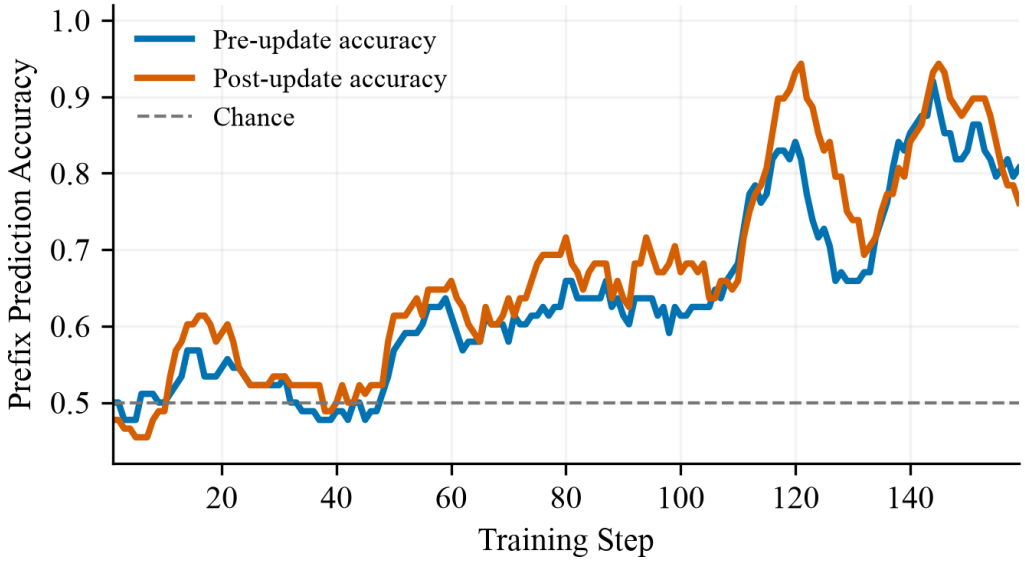
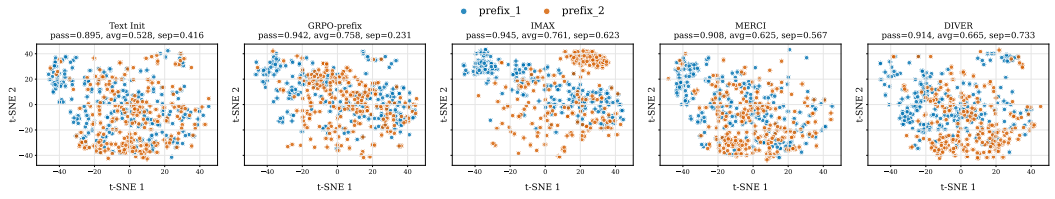


Figure 6: Prefix-identification accuracy of q_ϕ during IMAX training on Qwen3-8B with two prefixes. Pre-update and post-update accuracy are measured immediately before and after the posterior-head update at each training step; the dashed line denotes the 50% chance level.

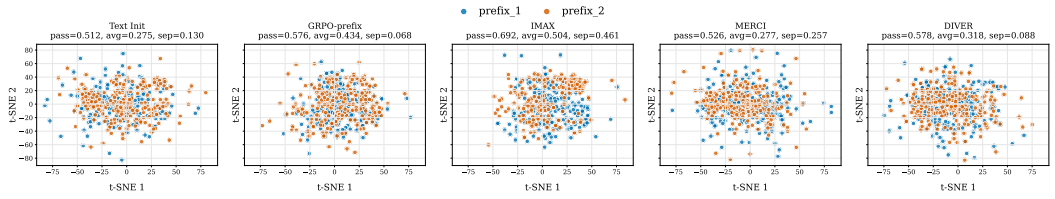
D Detailed Experiment Setup

D.1 Implementation Details

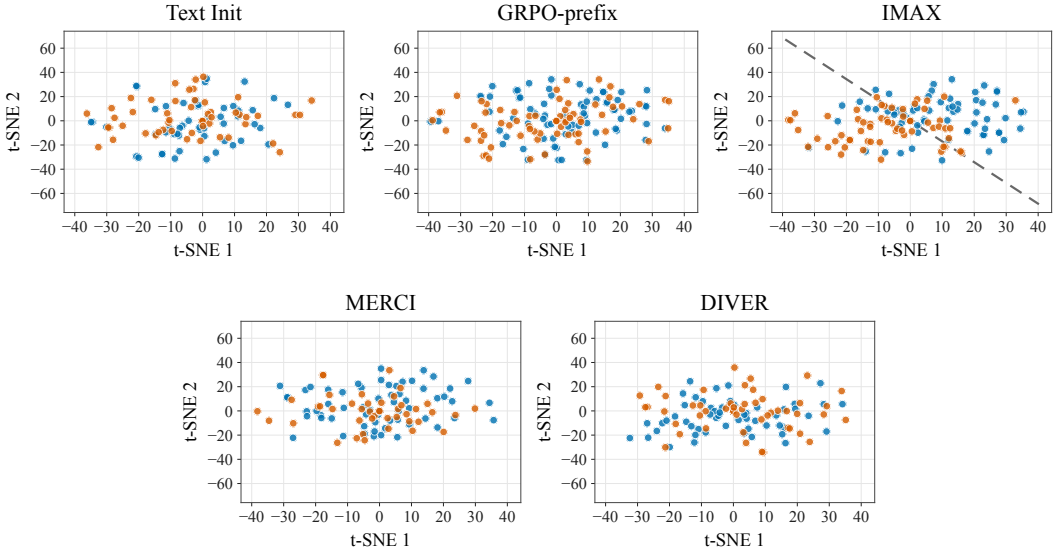
We train only the soft-prefix pool while keeping the backbone LLM frozen. For IMAX, the posterior model $q_\phi(z \mid x, y)$ is trained from prompt–response pairs using the frozen backbone representation and a trainable classification head; the resulting response-level InfoMax reward is then added to the verifiable reward before the prefix update. The main training and decoding hyperparameters are summarized in Table 10.



(a) GSM8K.



(b) MATH-500.



(c) Minerva Math.

Figure 7: Complete t-SNE visualizations of response embeddings for the two trained prefixes across GSM8K, MATH-500, and Minerva Math. Blue and orange points denote Prefix 1 and Prefix 2, respectively.

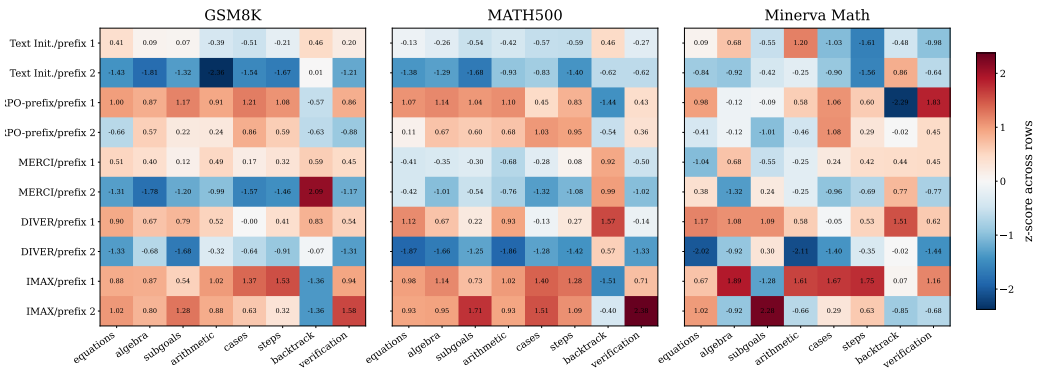


Figure 8: Reasoning-component contrast between Prefix 1 and Prefix 2 on GSM8K, MATH-500 and Minerva MATH.

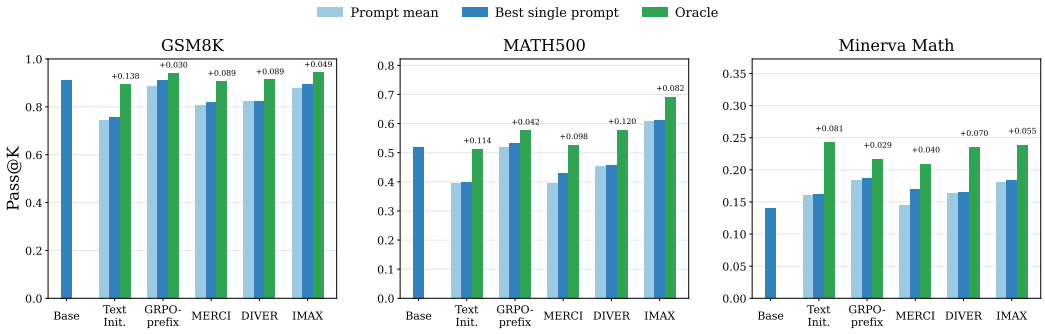
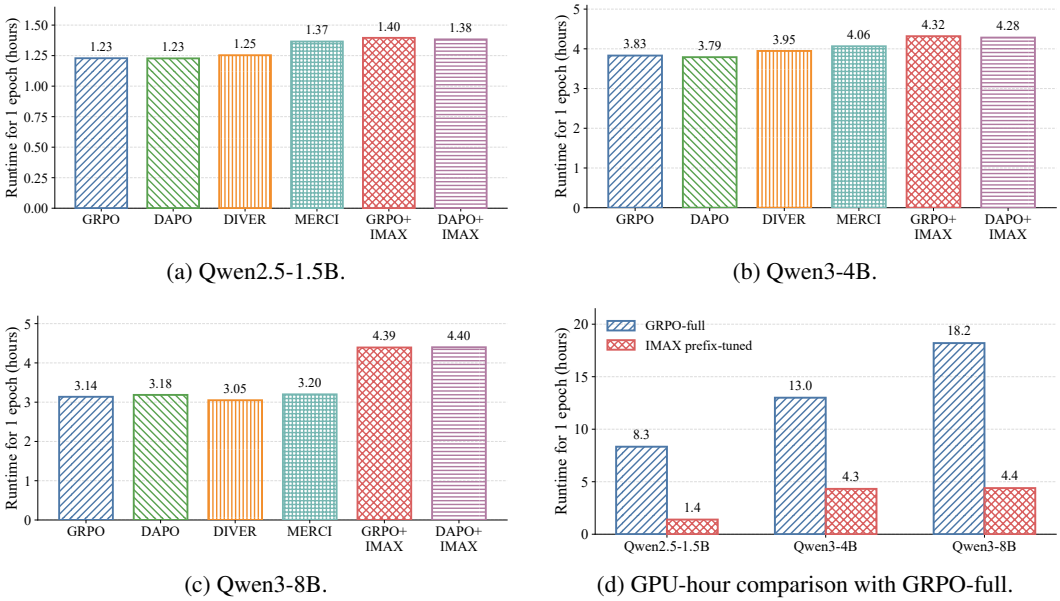


Figure 9: Complementary correctness analysis for the two prefixes. Prompt mean and best single prompt measure per-prefix solution quality. Oracle marks an example correct when either prefix succeeds, and the annotated oracle–best gap measures additional correct coverage contributed by the other prefix.

Table 8: Prefix-pool ablation on Qwen2.5-1.5B-Instruct under a matched total training-trajectory budget.

Run	Metric	GSM8K	MATH	Minerva	GPQA	AMC	AIME24	AIME25	IFEval
SFT	Pass@4	83.40±1.00	54.80±2.20	14.70±2.20	66.20±3.40	68.80±1.30	3.30±3.30	0.00±0.00	54.70±2.10
SFT	Avg@4	58.40±1.00	28.50±1.40	6.00±1.00	27.10±1.80	29.90±0.80	0.80±0.80	0.00±0.00	35.80±1.70
IMAX-1 prefix	Pass@4	83.60±1.00	58.60±2.20	18.80±2.40	66.20±3.40	69.30±1.30	0.00±0.00	0.00±0.00	54.50±2.10
IMAX-1 prefix	Avg@4	64.30±1.10	40.30±1.80	9.00±1.40	27.30±1.90	34.90±0.90	0.00±0.00	0.00±0.00	39.10±1.80
IMAX-2 prefixes	Pass@4	85.75±1.17	60.00±2.19	18.38±2.35	62.12±3.46	68.06±1.35	0.00±0.00	3.33±3.33	53.05±2.15
IMAX-2 prefixes	Avg@4	67.40±1.22	42.30±1.86	9.47±1.43	26.77±1.97	34.45±0.95	0.00±0.00	0.83±0.83	36.78±1.75
IMAX-4 prefixes	Pass@4	86.60±0.90	58.40±2.20	19.50±2.40	61.10±3.50	67.60±1.40	6.70±4.60	3.30±3.30	54.70±2.10
IMAX-4 prefixes	Avg@4	68.20±1.00	41.60±1.90	9.50±1.40	26.90±2.00	34.90±1.00	1.70±1.20	0.80±0.80	37.60±1.70



(a) Qwen2.5-1.5B.

(b) Qwen3-4B.

(c) Qwen3-8B.

(d) GPU-hour comparison with GRPO-full.

Figure 10: Runtime analysis for prefix-tuned methods and full-parameter GRPO. The first three panels report one-epoch runtime for each method under the prefix-tuning setting. The last panel compares IMAX with full-parameter GRPO using GPU-hours to account for the different number of GPUs used in training.

D.2 Default Prefix Pool

The default text prompt pool used to initialize the soft-prefix pool is shown in Table 9.

For evaluation, we use stratified prefix assignment and sample $N_{\text{eval}} = 2$ responses for each of two prompt–prefix pairs, yielding four responses per problem unless otherwise specified. We report Pass@4 and Avg@4 on reasoning benchmarks and IFEval, matching the metrics used in the main text. The full result tables in Appendix C include standard errors for both metrics.

E Dataset and Code Licenses

We list the datasets and codebases used in this work in Table 11. We report the license declared by the official release when available. For contest-derived evaluation sets whose public benchmark releases do not state an explicit license, we mark the license as not specified and use them only for evaluation.

Table 9: Default text prompt pool for soft-prefix initialization.

ID	Text Prompt
1	Reasoning Strategy: Introduce variables explicitly, write down the governing equations or identities, and solve through exact symbolic manipulation. Prefer precise derivation over heuristic guessing.
2	Reasoning Strategy: Partition the problem into exhaustive, mutually exclusive cases. Solve each case carefully, rule out impossible ones, and combine the valid conclusions.
3	Reasoning Strategy: Transform the problem into an equivalent but simpler form. Use substitution, reparameterization, factorization, or an alternative representation if it exposes the structure.
4	Reasoning Strategy: Look for an invariant, symmetry, conserved quantity, or monotonic property. Re-express the problem around that structure before solving.
5	Reasoning Strategy: Examine small, boundary, or special cases to discover a pattern or conjecture. Then generalize it and verify it rigorously for the full problem.
6	Reasoning Strategy: List the strongest constraints first and use them to eliminate impossible values, configurations, or structures before detailed computation.
7	Reasoning Strategy: Construct the required object, value, or configuration explicitly when possible, and then prove that the construction satisfies all conditions.
8	Reasoning Strategy: Focus on an extreme case, maximal or minimal element, or assume the opposite of the target claim and derive a contradiction to force the solution.

Table 10: Training and decoding settings used for prefix-tuned RLVR experiments.

Settings	Value
Trainable parameters	Soft prefixes and q_ϕ head
Backbone LLM	Frozen
Num virtual tokens	32
Max steps	300
Max response length	2048
Learning rate	5×10^{-4}
q_ϕ learning rate	1×10^{-4}
Intrinsic reward weight	0.01
Per-device train batch size	4
Per-device eval batch size	8
Gradient accumulation steps	8
Training temperature	1.0
Prefix assignment	Stratified
Number of prefixes C	2
Group size per prompt–prefix pair N	4
Evaluation rollouts per prompt–prefix pair N_{eval}	2
Evaluation temperature	0.7
Evaluation top-k	20
Evaluation top-p	0.8

Table 11: Licenses of datasets and codebases used in this paper.

Resource	Use in this paper	License	Source / notes
MATH (Hendrycks et al., 2021)	Training and evaluation	MIT	Official MATH dataset release: https://github.com/hendrycks/math.
MATH-500 (Lightman et al., 2023)	Evaluation	MIT	MATH-500 is the held-out MATH split from PRM800K: https://github.com/openai/prm800k.
GSM8K (Cobbe et al., 2021)	Evaluation	MIT	Official GSM8K release: https://github.com/openai/grade-school-math.
Minerva Math (Lewkowycz et al., 2022)	Evaluation	CC BY-NC-SA 4.0	Public Minerva Math benchmark mirror sourced primarily from MIT OpenCourseWare materials.
AMC (Hendrycks et al., 2021)	Evaluation	Apache-2.0	NuminaMath / AMC-derived release used by the evaluation suite.
AIME24	Evaluation	Apache-2.0	Math-AI AIME24 release: https://huggingface.co/datasets/math-ai/aime24.
AIME25	Evaluation	Apache-2.0	Math-AI AIME25 public benchmark release.
GPQA (Rein et al., 2024)	Evaluation	CC BY 4.0	Official GPQA release: https://huggingface.co/datasets/Idavidrein/gpqa.
IFEval (Zhou et al., 2023)	Evaluation	Apache-2.0	Official IFEval release: https://huggingface.co/datasets/google/IFEval.
TRL	Training implementation	Apache-2.0	Hugging Face TRL codebase: https://github.com/huggingface/trl.
lighteval	Evaluation implementation	MIT	Hugging Face lighteval codebase: https://github.com/huggingface/lighteval.