Pass@k Training for Adaptively Balancing Exploration and Exploitation of Large Reasoning Models

Anonymous ACL submission

Abstract

Reinforcement learning with verifiable rewards (RLVR) typically adopts Pass@1 as the reward, causing policies to prefer conservative and similar actions, resulting in issues in balancing exploration and exploitation. Thus, identifying an appropriate reward metric is crucial. Motivated by adapting the Pass@k metric as the reward to enhance exploration from previous work, we wonder whether this approach can be used to *balance the conflict between the model's exploration and exploitation abilities*. To investigate this, we first adopt Pass@k as the reward in the popular RLVR to train a policy (*i.e.,* **Pass@k Training**) in three variants. During analysis, we observe the improvement in its exploration ability without compromising its exploitation ability. Our further analysis reveals that the benefits from the improvement in exploration can be brought to exploitation, and these two abilities can mutually enhance each other. Following the findings, we explore the **implicit reward design** in RLVR, directly designing the advantage functions to enable better balance, yielding promising improvements on both Pass@1 and Pass@k scores, and highlighting a potential future research direction.

1 Introduction

Recently, reinforcement learning with verifiable rewards (RLVR) has boosted the reasoning ability of large language models (LLMs) (Trung et al., 2024; Lambert et al., 2024; Zhang et al., 2024). During RLVR, LLMs perform exploration based on the given prompt and learn from these trajectories through outcome-level rewards (Shao et al., 2024; Kimi et al., 2025; Guo et al., 2025). After that, LLMs achieve remarkable performance on complex tasks (Min et al., 2024; Chen et al., 2025b).

The current RLVR optimizes the Pass@1 objective, also known as *Pass@1 Training* (Xie et al., 2025; Hu et al., 2025a), leading to a major challenge of the balance of exploration and exploitation (Cheng et al., 2025; Hou et al., 2025). Typically, exploration refers to performing novel and various actions (Wang et al., 2024), while exploitation requires LLMs to invoke reliable actions that the verifier prefers among the known behaviours (Singh et al., 2024). Pass@1 Training leads LLM to focus more on exploitation, which may fail to explore trajectories with meaningful learning value in RLVR (Cui et al., 2025; Li et al., 2021), thus significantly affecting RLVR effectiveness and limit the upper bound of LLM ability.

The cause of this issues is that a LLM solution that contains sound reasoning paths but yields an wrong answer will be assigned negative rewards (Chen et al., 2024; Sun et al., 2025), discouraging it from employing similar methods in subsequent reasoning, thereby leading LLM adopting monotonous and conservative actions (Li et al., 2021; Amin et al., 2021). To mitigate this issue, we advocate for a new RLVR approach with a higher tolerance for incorrect responses, reducing the penalization of potentially correct actions. Fortunately, Pass@k metric has emerged to assess whether models can generate correct responses within k attempts, assessing LLM ability boundaries (Brown et al., 2024). Compared with the Pass@1 metric, the Pass@k metric allows the policy to generate several incorrect responses.

To employ the Pass@k metric as the supervision signal, prior work, *e.g.,* PKPO (Walder and Karkhanis, 2025), utilizes it as a reward function in traditional reinforcement learning to enhance policy exploration. To maximize the probability of generating at least one positive sample, a "smart" policy will generate k diverse solutions to cover the possible solution space, rather than k similar samples. Actually, a higher Pass@k score does not necessarily translate into improved helpfulness of LLMs in real-world applications (Miller and Tang, 2025; Hariri et al., 2025). Thus, we ask a more practical and impactful question: *how can Pass@k*

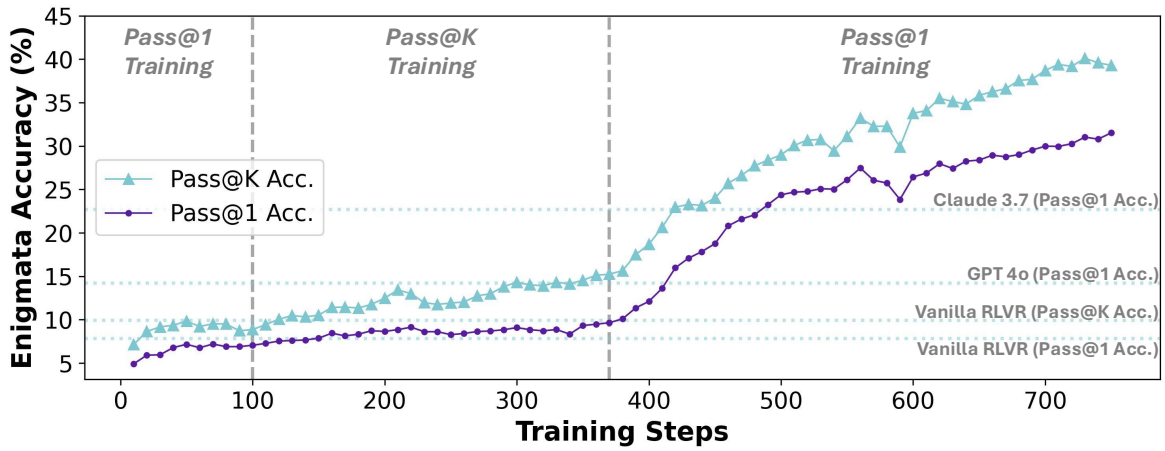


Figure 1: The performance of Qwen2.5-7B-Ins on Enigmata (Chen et al., 2025a), which is a puzzle task (Appendix B.1). **Pass@k Training** activates the exploration ability, surpassing vanilla RLVR and powerful LLMs.

metric be leveraged in current RLVR algorithms to balance LLM exploration and exploitation, thereby boosting both the Pass@1 and Pass@k scores?

To answer this question, we develop the necessary advantage estimation methods to integrate the Pass@k metric as the reward in the popular RLVR (*i.e.,* DAPO (Yu et al., 2025)) in three variants (**Pass@k Training**). We find that the trained model can achieve higher Pass@k score and comparable Pass@1 score than vanilla RLVR. To understand its mechanism, we study how Pass@k Training balances exploration and exploitation in LLMs and what the benefits it brings. In particular, we observe the improvements in Pass@1 score, which is a key metric in real-world applications, and find that an appropriate advantage estimation method can further enhance specific LLM performance. Following this insight, we explore how to enhance the RLVR via **implicit reward design** to adaptively balance exploration and exploitation, and avoid complex derivation, showing the great enhancement on both Pass@1 and Pass@k score. The vital takeaways can be summarized as follows:

- Pass@k Training significantly enhances the exploration ability of LLMs, and also improves Pass@1 scores. Among its three variants, bootstrap sampling offers higher efficiency than full sampling, and analytical derivation mitigates the variance introduced by sampling. (Sec. 3)
- Pass@k Training is robust and effective, and its enhancement of LLM exploration is beneficial to their exploitation through further Pass@1 Training, leading 7B LLM to surpass the powerful LLMs, highlighting its practical value. (Sec. 4)
- To improve both Pass@1 and Pass@k scores of LLMs, we observe that the implicit reward design allows finer control over optimization, without complex derivations. It is a promising direction for future RLVR development. (Sec. 5)

2 Related Work

In this section, we briefly introduce the related work, and the details can be found in Appendix A.

RLVR for LLMs. Reinforcement learning with verifiable rewards (RLVR) can unleash the potential of LLM reasoning ability (DeepSeek-AI et al., 2025; Chen et al., 2025b; Zeng et al., 2025; Hu et al., 2025a) the RLVR training paradigm can help LLMs to control their reasoning time (Aggarwal and Welleck, 2025), switch the reasoning pattern (Chung et al., 2025; Wu et al., 2025), and enhance the specific performance metric (Tang et al., 2025). Specifically, previous work (Walder and Karkhanis, 2025) utilizes Pass@k as the reward on the policy gradient (Williams, 1992) to encourage models to solve hard problems. However, the intrinsic connection between Pass@k RLVR training and LLM exploration ability has not been fully recognized. Thus, we further adopt the Pass@k metric in GRPO and its variants through three approaches.

Exploration in Test-time. Recently, test-time scaling consumes more computational resources at inference time to enhance the performance of LLMs (Zhang et al., 2025). Effective exploration is important and necessary during the test-time scaling process (Hou et al., 2025; Luo et al., 2025). However, existing work reveals that the exploration ability is limited by the corresponding base model (Yue et al., 2025). To mitigate this issue,

previous work proposed several approaches, including achieved by adjusting the sampling hyperparameters (Chen et al., 2025b; Hu et al., 2025a; Hou et al., 2025), performing self-verification and self-reflection (Jiang et al., 2025; Liu et al., 2025b; Sareen et al., 2025), or leveraging external models to verify the reasoning process (Liu et al., 2025d; Zha et al., 2025). In this work, we explore whether better supervision signals can be designed to balance the model's exploration and exploitation.

3 Pass@k as Reward in RLVR Training

In this section, we first formulate the reasoning tasks and Pass@1 Training (Sec. 3.1). Next, we introduce how to implement Pass@k as a reward in RLVR (Sec. 3.2), and then propose two enhancements (Sec. 3.3 and Sec. 3.4). Finally, we conduct the experiments to analysis the advantages of these three variants (Sec. 3.5). We present a demonstration in Fig. 2, and provide the details of the experiments and benchmarks in Appendix B.

3.1 Formulation and Preliminary

Typically, a reasoning problem from the dataset $\mathcal{D}$ contains a question x and an answer y. A policy π_θ (*i.e.,* LLM with the parameters θ) needs to generate a response $\hat{y} = \{t_1, \ldots, t_l\}$ based on x, where t_i and l refer to the i-th token and the length of $\hat{y}$. A verifier provides a reward $R(y, \hat{y}) \in \{R_{\text{neg}}, R_{\text{pos}}\}$, where R_{neg} and R_{pos} are for negative and positive responses. Simplify, we use R to refer $R(y, \hat{y})$.

Based on the formulation, in Pass@1 Training (*e.g.,* GRPO (Shao et al., 2024)), the advantage is estimated through the average value and standard deviation of the rewards in same group as follows,

$$\bar{R} = \frac{1}{N_{\text{rollout}}} \sum_{i=1}^{N_{\text{rollout}}} R_i, \quad \sigma = \frac{1}{N_{\text{rollout}}} \sqrt{\sum_{i=1}^{N_{\text{rollout}}} (R_i - \bar{R})^2}, \quad \hat{A}_{i,1} = \cdots = \hat{A}_{i,|\hat{y}_i|} = \frac{R_i - \bar{R}}{\sigma}, \tag{1}$$

where N_{rollout} refers the number of the responses, and R_i refer to the rewards of i-th response. GRPO utilizes the following equation to compute the objective function $\mathcal{J}(\theta)$ to optimize the policy,

$$\mathcal{J}(\theta) = \mathbb{E}_{(q,a)\sim\mathcal{D},\{o_i\}_{i=1}^{G}\sim\pi_\theta(\cdot|q)} \Big[\frac{1}{G} \sum_{i=1}^{G} \frac{1}{|\hat{y}_i|} \sum_{t=1}^{|\hat{y}_i|} \Big(\min \Big(r_{i,t} \hat{A}_{i,t}, \text{clip}\left(r_{i,t}, 1-\varepsilon, 1+\varepsilon\right) \hat{A}_{i,t} \Big) - \beta D_{\text{KL}} \Big) \Big]. \tag{2}$$

Since each token shares the same advantage in GRPO, we will not distinguish at the token level in the following and use $\hat{A}_i$ to refer to the advantage of the i-th response. In practice, we employ a variant of GRPO, *i.e.,* DAPO (Yu et al., 2025).

3.2 Pass@k Training via Full Sampling

The behaviour of LLMs can be adjusted by the specific rewards (Tang et al., 2025). Motivated by it, since the Pass@k metric can reflect LLM exploration ability, we first formulate the Pass@k metric and then integrate it into RLVR.

Fromulation of Pass@k Metric. Given a question x, the policy model generates k responses through a specific strategy (*e.g.,* sampling-based decoding or Monte Carlo Tree Search). The i-th sampled response $\hat{y}_i$ will receive a reward R_i, which is provided by the verifier. Based on this, the Pass@k score is defined as the expected maximum reward obtained from these k responses. Formally, the Pass@k metric can be computed as follows,

$$\text{Pass@k} = \mathbb{E}_{(x,y)\sim D, \{\hat{y}_i\}_{i=1}^{k}\sim\pi_\theta(\cdot|x)} \left[\max\left(R_1, \ldots, R_k\right)\right]. \tag{3}$$

Simple Full Sampling. First, we rollout the responses $\hat{\mathcal{Y}} = \{\hat{y}_1, \ldots, \hat{y}_{N_{\text{rollout}}}\}$ from π_θ for the given question and separate these responses into $N^{\text{group}} = \lfloor \frac{N_{\text{rollout}}}{k} \rfloor$ groups (discarding the redundancy), where the j-th group contains the k responses $\hat{\mathcal{Y}}^j = \{\hat{y}_{k\times(j-1)+1}, \ldots, \hat{y}_{k\times(j-1)+k}\}$. Next, the group reward is computed by the maximum reward of the responses it contains. The advantage of j-th group $\hat{A}^j$ can be calculated following the DAPO advantage estimation approach. After that, we divide the group advantage to the responses it contains, *i.e.,* $\hat{A}_{k\times(j-1)+1} = \cdots = \hat{A}_{k\times(j-1)+k} = \hat{A}^j$. Finally, we can utilize these information to optimize the policy.

3.3 Bootstrap Sampling for Efficiency

The rollout times of Full Sampling increase significantly when N^{group} increases, costing more resources. However, an insufficient N^{group} can easily introduce noise due to sampling. Thus, we utilize bootstrap sampling to reduce rollout times while maintaining N^{group}, alleviating the affect of noise.

Bootstrap Sampling. We first rollout the responses $\hat{\mathcal{Y}} = \{\hat{y}_1, \ldots, \hat{y}_{N_{\text{rollout}}}\}$ from π_θ. To construct N^{group} groups, we randomly sample k responses from $\hat{\mathcal{Y}}$, and these sampled responses constitute a group. Concretely, to construct the j-th group, we select k distinct indices of the generated responses $\mathcal{P} = \{p_{j,1}, \ldots, p_{j,k}\}$. The cor-

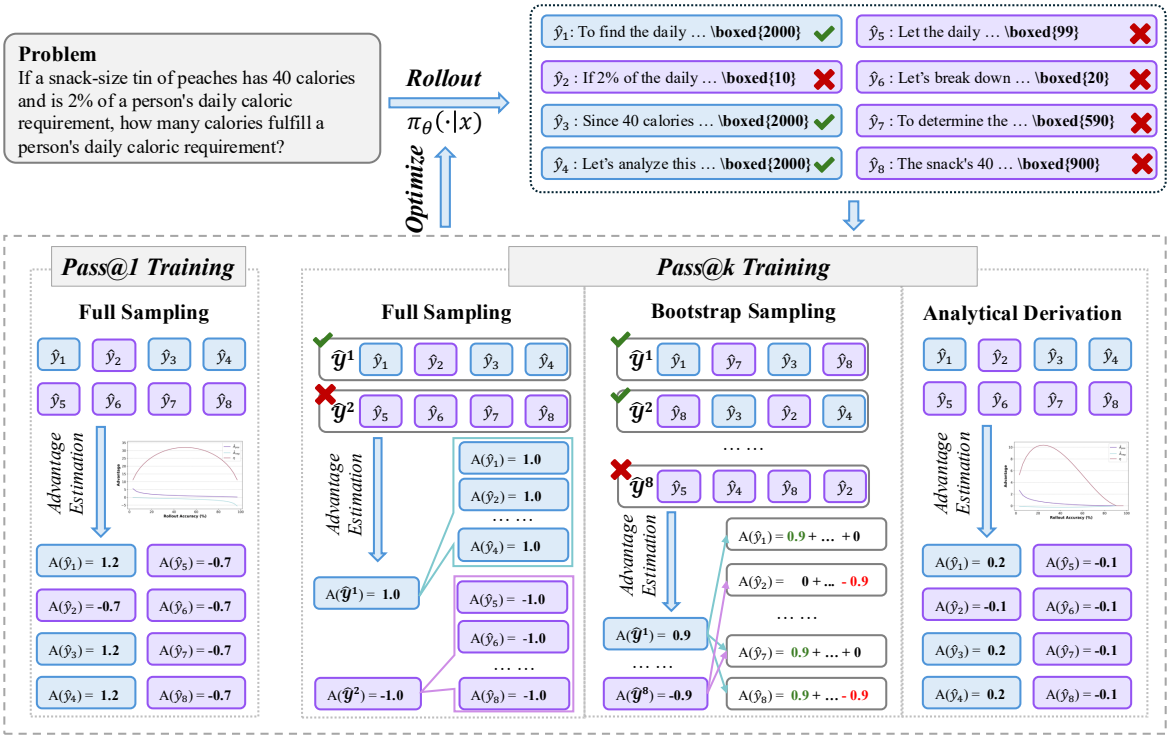


Figure 2: The comparison between Pass@1 and Pass@k Training. Full sampling, bootstrap sampling, and analytical derivation are three progressive enhancements for Pass@k Training. We present the pseudo-code in Appendix E.

responding responses construct the group $\hat{\mathcal{Y}}^j = \{\hat{y}_{p_{j,1}}, \ldots, \hat{y}_{p_{j,k}}\}$. We repeat this process N^{group} times to collect N^{group} groups. Next, we estimate the group advantage with a method similar to Full Sampling. Since responses may appear in multiple groups by bootstrap sampling, we average advantages of the groups it belongs to as its advantage,

$$\hat{A}_i = \frac{1}{N^{\text{group}}} \sum_{j=1}^{N^{\text{group}}} \hat{A}^j \cdot \mathbb{I}[\hat{y}_i \in \hat{\mathcal{Y}}^j], \tag{4}$$

where $\mathbb{I}[\hat{y}_i \in \hat{\mathcal{Y}}^j]$ returns 1 if and only if $\hat{y}_i$ belongs to the j-th group $\hat{\mathcal{Y}}^j$, while returns 0 for others.

3.4 Analytical Derivation for Effective and Efficient Pass@k Training

Following the idea of bootstrap sampling, we derive the response advantage (*i.e.,* $\hat{A}_{pos}$ and $\hat{A}_{neg}$) to remove the variance from the group construction. Detailed derivations are in Appendix D.

Analytical Derivation. A group that contains at least one positive response (*positive group*) will be assigned the positive reward R_{pos}, while the other groups (*negative group*) will be endowed with the negative reward R_{neg}. We start by analyzing the average reward and standard deviation of the groups, *i.e.,* $\bar{R}^{\text{group}}$ and σ^{group}. First, the average reward of the group can be formulated:

$$\bar{R}^{\text{group}} = \frac{1}{N^{\text{group}}_{\text{total}}} \times \left(N^{\text{group}}_{\text{pos}} \times R_{\text{pos}} + N^{\text{group}}_{\text{neg}} \times R_{\text{neg}} \right), \tag{5}$$

where $N^{\text{group}}_{\text{total}}$ refers to the total number of groups, and $N^{\text{group}}_{\text{pos}}$ and $N^{\text{group}}_{\text{neg}}$ denote the number of positive and negative groups, respectively. As each group is constructed by selecting k responses, we can obtain that $N^{\text{group}}_{\text{total}} = \binom{N_{\text{rollout}}}{k}$. Considering the negative group, when and only when randomly sampling k negative responses from $\hat{\mathcal{Y}}$, these sampled responses can construct a negative group. Thus, the number of negative and positive groups can be calculated, *i.e.,* $N^{\text{group}}_{\text{neg}} = \binom{N_{\text{neg}}}{k}$ and $N^{\text{group}}_{\text{pos}} = \binom{N_{\text{rollout}}}{k} - \binom{N_{\text{neg}}}{k}$. Based these, we can obtain $\bar{R}^{\text{group}}$ and σ^{group} as follows,

$$\bar{R}^{\text{group}} = 1 - \frac{\binom{N_{\text{neg}}}{k}}{\binom{N_{\text{rollout}}}{k}}, \quad \sigma^{\text{group}} = \sqrt{\bar{R}^{\text{group}} \times \left(1 - \bar{R}^{\text{group}}\right)}. \tag{6}$$

Then, we can deduce the group advantage (*i.e.,* $\hat{A}^{\text{group}}_{\text{pos}}$ and $\hat{A}^{\text{group}}_{\text{neg}}$) as follows,

$$\hat{A}^{\text{group}}_{\text{pos}} = \frac{1 - \bar{R}^{\text{group}}}{\sigma^{\text{group}}}, \quad \hat{A}^{\text{group}}_{\text{neg}} = -\frac{\bar{R}^{\text{group}}}{\sigma^{\text{group}}}. \tag{7}$$

For $\hat{A}_{pos}$ and $\hat{A}_{neg}$, we need to consider the correctness of the group each response belongs to. Typically, a response will belong to $\binom{N_{\text{rollout}}-1}{k-1}$ groups,

because a group can be formed with the current response and other $k-1$ responses selected from the remaining $N_{\text{rollout}}-1$ responses. For a positive response, the group it belongs to is always the positive group. Thus, *the advantage of a positive response* $\hat{A}_{pos}$ can be calculated as follows,

$$\hat{A}_{\text{pos}} = \left(1 - \bar{R}^{\text{group}}\right) \times \left(\sigma^{\text{group}}\right)^{-1}. \tag{8}$$

For a negative response, it belongs to a negative group if and only if with other $k-1$ negative responses. Thus, the number of groups is $\binom{N_{\text{neg}}-1}{k-1}$. We can compute the number of positive groups by subtracting the number of negative groups from the total, *i.e.,* $\binom{N_{\text{rollout}}-1}{k-1} - \binom{N_{\text{neg}}-1}{k-1}$. Thus, *the advantage of a negative response* $\hat{A}_{neg}$ is as follows,

$$\hat{A}_{\text{neg}} = \left(1 - \bar{R}^{\text{group}} - \frac{\binom{N_{\text{neg}}-1}{k-1}}{\binom{N_{\text{rollout}}-1}{k-1}}\right) \times \left(\sigma^{\text{group}}\right)^{-1}. \tag{9}$$

We directly employ the analytical solutions of $\hat{A}_{\text{pos}}$ and $\hat{A}_{\text{neg}}$ in the advantage estimation of RLVR.

Discussion of $\hat{A}_{\text{pos}}$ and $\hat{A}_{\text{neg}}$. First, we prove that *Analytical Derivation exhibits lower bias and variance than Full Sampling* (Appendix D.6). Second, $\hat{A}_{\text{pos}} \geq 0$ and $\hat{A}_{\text{neg}} \leq 0$ consistently, reinforcing positive responses and penalizing negative ones. Analytical Derivation improves accuracy by eliminating noise. Third, compared to Pass@1 Training, Pass@k Training emphasizes hard problems, boosting the model's exploration ability (Appendix H). Finally, in Pass@k Training, we find that directly modifying the advantage function can improve the corresponding metrics. The explicit design of advantage can be regarded as the **implicit reward design**. We do not focus on the specific form of the reward function. Rather, our attention is directed toward the training objective. In Sec. 5, we explore it and find that it can integrate Pass@k Training with other methods to achieve improvement, highlighting it as a promising research direction.

3.5 Empirical Insights of Pass@k Training

We evaluate three variants of Pass@k Training against the baselines in Table 1.

Full Sampling Improves LLM Exploration. We express the training progress of Full Sampling and Pass@1 Training in Fig. 7. We observe that Pass@k Training can effectively improve the model's Pass@k score while also leading to improvements in the Pass@1 score, indicating that Pass@k Training does not introduce a conflict between exploration and exploitation. Moreover, as the N_{rollout} increases, the performance is further improved, indicating that Full Sampling suffers from high variance issues that need to be mitigated.

Table 1: Performance on Maze task on Qwen2.5-7B-Ins of Pass@1 Training and the three variants of Pass@k Training in the setting of $k = 4$.

Methods	Pass@1 Score	Pass@k Score
Qwen2.5-7B-Ins	18.3	25.0
Pass@1 Training	32.4	33.0
Pass@k Training via Full Sampling		
w/ $N_{\text{rollout}} = 32$	26.4	47.6
w/ $N_{\text{rollout}} = 128$	28.9	53.1
Pass@k Training via Bootstrap Sampling		
w/ $N_{\text{rollout}} = 32$	37.8	51.0
Pass@k Training via Analytical Derivation		
w/ $N_{\text{rollout}} = 32$	94.6	100.0

Bootstrap Sampling Reduces Training Budget. In Fig. 8 and Table 1, given the same rollout number (*i.e.,* $N_{\text{rollout}} = 32$), bootstrap sampling outperforms full sampling. This improvement arises from the fact that bootstrap sampling generates a larger number of groups, reducing the variance of the sampled reward distribution to the true distribution, leading to more stable training. Additionally, it achieves comparable performance to Pass@1 Training on the Pass@1 scores.

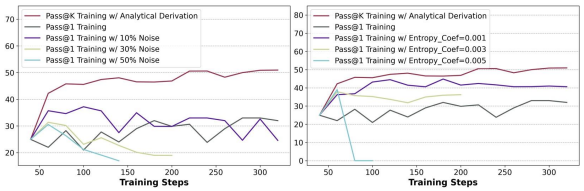
Analytical Derivation Further Improves Exploration. We assess Pass@k Training on different LLMs on various tasks in Appendix G. In Fig. 9 and Table 1, we observe that Pass@k Training achieves a higher Pass@k score than Pass@1 Training, showing its effectiveness. When the training steps increase, Bootstrap Sampling exhibits performance fluctuations due to instability. In contrast, since Analytical Derivation eliminates the sampling for group construction, it provides a more stable RLVR and leads to continuous performance improvements. Besides, it also enhances the Pass@1 score of LLM, surpassing Pass@1 Training.

4 Inner Mechanism of Pass@k Training

First, we analyze how Pass@k Training affects LLM exploration (Sec. 4.1) and examine the diversity and the entropy of LLM responses to evaluate its exploration ability (Sec. 4.2). Moreover, we analyze the effect of value k to assess the robustness of Pass@k Training (Sec. 4.3). Finally, we ex-

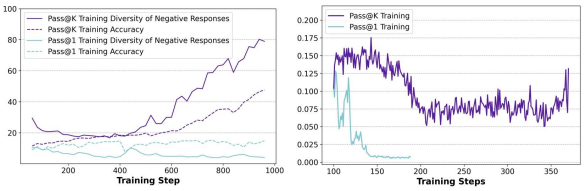
Table 2: Pass@1/Pass@k performance of Enigmata of different methods. "P@1 T." and "P@k T." denote the Pass@1 Training and Pass@k Training with analytical derivation based on DAPO.

	Crypto	Arithmetic	Logic	Grid	Graph	Search	Sequential	Overall
			Closed-source LLMs w/o RLVR Training (Pass@1)					
GPT-4o-1120	26.2	1.9	34.5	17.8	19.3	6.0	3.9	14.2
Claude-3.7-Sonnet	38.1	16.7	60.0	22.9	22.4	7.8	15.0	22.7
Qwen2.5-7B-Ins	0.1/0.7	1.0/3.3	28.1/48.4	3.7/9.2	3.0/11.6	0.3/1.0	2.7/5.4	4.7/10.1
+ PKPO	0.0/0.0	1.6/8.0	32.4/43.6	11.5/20.5	9.8/12.7	3.7/8.1	8.9/10.2	10.1/15.4
+ ForkingToken	78.9/94.0	10.7/35.0	46.2/67.1	19.6/25.9	14.6/21.6	7.4/13.4	10.0/11.7	20.6/29.1
+ EntropyAdv	46.7/73.3	31.3/69.3	48.6/72.7	24.4/31.5	17.0/31.5	8.8/19.5	16.2/18.7	23.3/35.8
+ P@1 T.	1.2/5.7	6.6/28.0	41.1/68.4	14.8/22.8	14.5/20.4	3.3/6.8	9.6/12.7	12.9/21.3
+ P@k T.	14.0/39.7	27.4/63.0	46.0/74.0	18.8/27.8	15.2/21.5	5.3/12.3	14.1/18.3	17.9/29.8
+ P@k T. + P@1 T.	96.9/98.3	36.2/67.7	49.3/71.8	30.9/37.5	20.3/30.7	25.8/37.5	10.6/12.9	**30.8/40.6**
Qwen2.5-32B-Ins	1.5/5.3	4.6/16.0	45.7/71.1	11.6/20.6	8.0/26.7	2.3/6.4	7.7/16.3	10.9/21.6
+ P@1 T.	95.8/99.7	53.0/85.0	76.6/92.2	38.4/47.4	44.4/57.8	47.0/58.8	21.8/25.8	45.2/56.0
+ P@k T.	93.8/99.3	51.1/86.3	74.8/92.4	39.0/49.6	42.7/61.3	45.9/59.9	21.5/26.6	44.5/57.4
+ P@k T. + P@1 T.	95.9/99.3	49.6/84.3	82.0/94.9	40.0/51.0	48.2/60.2	48.8/60.8	22.2/26.2	**46.8/57.9**



(a) Noise Reward. (b) Entropy Reg.

Figure 3: Pass@k scores of Maze tasks in RLVR.



(a) Answer Diversity. (b) Entropy of Policy.

Figure 4: Results on Maze task in RLVR.

plore how to transfer the benefits to Pass@1 scores (Sec. 4.4). More discussions are in Appendix I.

4.1 How does Pass@k Training Compare to the Variants of Pass@1 Training?

Motivated by the prior work (Hou et al., 2025), we compare Pass@k Training with two baselines, *i.e.,* Noise Rewards and Entropy Regularization.

Noise Rewards. In Sec. 3.2, a negative response may receive the positive reward if it belongs to a positive group, raising the question of whether the noise rewards partially drive the improvement of Pass@k Training. To investigate this, we experiment in which a certain ratio (*i.e.,* 10%, 30%, and 50%) of negative responses are assigned the positive rewards. In Fig. 3a, guiding LLMs to learn from negative responses does not contribute to their reasoning ability, and a higher ratio of noise or additional training steps degrades model ability, showing that naively integrating noise into the reward cannot enhance exploration ability.

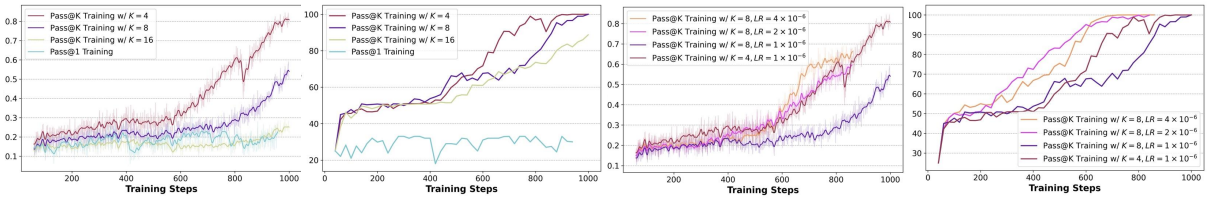
Entropy Regularization. Prior work (Cui et al., 2025) has found that entropy can be incorporated into RLVR to preserve the exploration ability of LLMs. Following this, we employ entropy regularization with the coefficient of $\{0.001, 0.003, 0.005\}$ in the RLVR process (Fig. 3b). We can find that a high coefficient might cause the collapse of the model. Although the small coefficient does not make LLM crush, it still underperforms Pass@k Training, and even leads to a performance decrease when the training steps increase, indicating that entropy regularization might affect training stability. Moreover, we discuss other entropy-guided methods in Appendix I.5.

Takeaway from Sec. 4.1. Simply adding noise to the rewards or introducing regularization terms into the objective function cannot stably and effectively improve the model's exploration capability.

4.2 Does Pass@k Training Elicit Exploration?

To analyze the LLM exploration in RLVR, we conduct an empirical study from the perspective of answer diversity and entropy of policy distribution.

Answer Diversity of Negative Responses. We counter the accuracy and the ratio of different answers of the negative responses of Pass@k and Pass@1 Training, aiming to assess the exploration ability of LLMs on the uncertain answer. Fig. 4a



(a) Training Rewards. (b) Pass@k Scores. (c) Training Rewards. (d) Pass@k Scores.

Figure 5: Training progress of Qwen2.5-7B-Ins on Maze Tasks of Pass@k Training under various k and learning rate (LR). In (a) and (b), LR = 1×10^{-6}.

Table 3: Pass@1/Pass@k Performance on mathematical tasks of different methods. "P@1 T." and "P@k T." denote the Pass@1 and Pass@k Training with analytical derivation. "×2" refers to repeating the process twice.

	AIME 2024	AIME 2025	OlymMATH-Easy	OlymMATH-Hard	Avg.
DeepSeek-R1-Distill-Qwen-1.5B	22.7/61.4	20.5/37.2	6.6/36.7	0.6/5.2	12.6/35.1
+ P@1 T.	36.7/76.0	28.8/49.4	16.7/51.7	2.5/17.5	21.2/48.7
+ P@k T.	36.5/79.3	27.0/55.5	17.6/59.3	2.4/17.4	20.9/52.9
+ P@k T. + P@1 T.	42.3/71.7	30.4/57.8	20.7/60.5	3.4/18.9	24.2/52.2
+ (P@k T. + P@1 T.) × 2	44.2/77.2	31.5/57.7	22.6/62.7	4.4/21.2	**25.7/54.7**

shows that the answer diversity of the negative response stays at the same level during Pass@1 training, indicating that the LLMs try to select the "safe" actions and tend to generate similar answers, limiting the scope of exploration. Differently, in Pass@k Training, LLM is encouraged to achieve a higher Pass@k score and naturally learn to generate diverse answers when it is not confident enough. Thus, the exploration ability of LLM is enhanced and thereby improves its exploitation ability.

Policy Entropy. In Fig. 4b, Pass@k Training keeps the entropy of policy distribution at a relatively high level, while Pass@1 Training induces entropy to converge to a low value. LLMs can keep their exploration ability during Pass@k Training, but will lose it during Pass@1 Training. Besides, we observe that Pass@k Training leads to an increase in entropy from around 200 steps, validating our hypothesis that using Pass@k as the training objective can encourage the model to conduct more exploration, thereby naturally increasing entropy.

Takeaway from Sec. 4.2. Pass@k Training encourages exploration only on low-confidence questions, enabling mutual gains in exploration and exploitation abilities of LLMs.

4.3 What is the Affect of the Value k?

We adjust k in $\{4, 8, 16\}$ in the RLVR (Fig. 5a and Fig. 5b). For various k, the training rewards and test set Pass@k score can be improved to a high level, indicating k is not a vital factor for improving LLM exploration ability. However, the improvement slows down with the increase of k. Based on the analytical solutions (*i.e.,* Eq. 8 and Eq. 12), a larger value of k may cause more samples to be discarded, because they cannot construct negative groups, which results in a lower efficiency. We explore whether enlarging the learning rate (LR) can mitigate this issue. We employ LR in $\{1 \times 10^{-6}, 2 \times 10^{-6}, 4 \times 10^{-6}\}$ with $k = 8$ (Fig. 5c and Fig. 5d). The increase of LR brings higher efficiency. The setting $k = 8$ with LR= 4×10^{-6} even outperforms $k = 4$, showing that such issues can be addressed.

Takeaway from Sec. 4.3. Pass@k Training is robust to the choice of k, maintaining stable performance, while reduced optimization efficiency at larger k can be mitigated by increasing LR.

4.4 Can Pass@k Training Boost Exploitation?

To transfer the benefits from Pass@k to Pass@1 score, a native method is performing Pass@1 Training after Pass@k Training. We present the results of puzzle tasks in Table 2 and mathematical tasks in Table 3, and provide the significance test in Appendix G.1. Moreover, we experiment with other LLMs on various tasks in Appendix G. First, the Pass@1 Training after Pass@k Training can improve the exploitation of LLMs, achieving a higher Pass@1 score. We observe that even the 7B model can surpass the powerful closed-source LLMs, *e.g.,* GPT-4o and Claude-3.7-Sonnet, because Pass@k Training activates the exploration ability of LLMs, unleashing their potential in the subsequent RLVR.

Second, either the small-scale or large-scale LLMs (*e.g.,* 7B or 32B model) can benefit from Pass@k Training. Third, the domain of tasks also does not affect the transfer from LLM Pass@k scores to their Pass@1 scores. Besides, we compare the Pass@k Training with other competitive baseline approaches, *i.e.,* PKPO (Walder and Karkhanis, 2025), ForkingToken (Wang et al., 2025), and EntropyAdv (Cheng et al., 2025), which are proposed to balance the exploration and exploitation. In Table 2, Pass@k Training (*i.e.,* "+ P@k T. + P@1 T.") outperforms these baselines in both Pass@1 and Pass@k score, showing its effectiveness and the practical value for real-world applications.

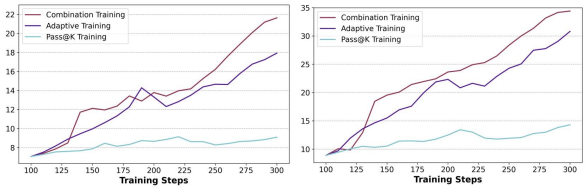
Takeaway from Sec. 4.4. Pass@k Training is consistently beneficial to the Pass@1 score across model scales and task domains.

5 Balancing Exploration and Exploitation

As discussed in Sec. 3.4, we find that appropriately designed advantage functions can enhance the specific LLM ability. Thus, we consider whether it is possible to design the advantage function (*i.e.,* **Implicit Reward Design**) beyond Pass@k Training to better balance LLM exploration and exploitation.

Combination Training: Combining Two Methods via Accuracy. We consider whether combining Pass@1 and Pass@k Training could be beneficial. Thus, we design the following formula to estimate the final advantage value: $\hat{A} = \frac{N_{\text{pos}}}{N} \times \hat{A}_{\text{Pass@k}} + (1 - \frac{N_{\text{pos}}}{N}) \times \hat{A}_{\text{Pass@1}}$, where $\hat{A}_{\text{Pass@k}}$ and $\hat{A}_{\text{Pass@k}}$ denote the advantages estimated by Pass@k and Pass@1 Training approach, respectively. When the sampled response has a low accuracy, the advantage value from Pass@1 Training receives a higher weight and dominates the training, leading to high training efficiency. Conversely, when the accuracy is high, the advantage of Pass@k Training is assigned a greater weight, avoiding LLMs from overfitting to easy problems.

Adaptive Training: Adaptive Selection via Entropy. The entropy of policy can indicate its exploration ability (Cui et al., 2025). Thus, we conduct the RLVR based on policy entropy. Concretely, we first compute the average entropy $\bar{E}$ of the sampled responses of each question, and then rank each problem based on its $\bar{E}$. We designate the top 50% as *high-exploration problems* and the rest as *low-exploration problems*. For high-exploration problems, we use the Pass@1 advantage function to help LLM exploit prior exploration, and apply the Pass@K advantage function to guide further exploration for others.



(a) Pass@1 Scores. (b) Pass@k Scores.

Figure 6: Pass@k Training and variants in RLVR.

Analysis and Discussion. We present the results of Qwen on Enigmata in Fig. 6, and observe that the variants outperform the standard Pass@k Training. Both Pass@1 and Pass@k scores improve rapidly and maintain a high growth rate during RLVR, indicating the complementarity of Pass@1 and Pass@K Training. By designing a proper method, it is possible to leverage the strengths of these methods, enabling LLM to achieve higher performance. In contrast, Pass@k Training leads to slower performance gains. This is because Pass@k Training encourages LLMs to explore, leading to a slow convergence rate. The analysis supports the idea that designing an advantage function based on the model's current state can effectively enhance its abilities. Implicit reward design is an extension of Pass@k Training, and it successfully balances exploration and exploitation of LLMs, highlighting a potential future direction.

6 Conclusion

In this work, we focused on how to balance the exploration and exploitation abilities of LLMs, and developed the necessary advantage estimation methods to integrate the Pass@k metric as the reward in the popular RLVR algorithms (**Pass@k Training**). According to our detailed analysis, we found that Pass@k Training can bring the benefits to Pass@1 scores, and an appropriate advantage estimation method is vital to enhance specific LLM performance, which leads to a key insight that designing appropriate advantage functions can better balance the exploration and exploitation abilities of LLMs. Following these insights, we investigated **implicit reward design**, which integrates Pass@k Training with various guidance without complex analytical derivation, showing remarkable improvement on both Pass@1 and Pass@k scores of LLMs, highlighting a promising future research direction.

Limitations

In this part, we present the discussion about the limitations of our work. First, for implicit reward design, we only conducted a preliminary exploration and did not investigate it in depth. In fact, our proposed method is already capable of alleviating the conflict between exploration and exploitation, effectively improving both the Pass@1 and Pass@k scores of the model. We leave further exploration to future work. Finally, we do not investigate broader ethical implications in this work. Nevertheless, since all experiments are conducted exclusively on scientific tasks, the potential ethical risks are minimal.

References

Pranjal Aggarwal and Sean Welleck. 2025. L1: controlling how long A reasoning model thinks with reinforcement learning. *CoRR*, abs/2503.04697.

AIME2024. 2024. Aime2024.

AIME2025. 2025. Aime2025.

Susan Amin, Maziar Gomrokchi, Harsh Satija, Herke van Hoof, and Doina Precup. 2021. A survey of exploration methods in reinforcement learning. *CoRR*, abs/2109.00157.

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

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, Tony Tong Wang, Samuel Marks, Charbel-Raphaël Ségerie, Micah Carroll, Andi Peng, Phillip J. K. Christoffersen, Mehul Damani, Stewart Slocum, Usman Anwar, and 13 others. 2023. Open problems and fundamental limitations of reinforcement learning from human feedback. *Trans. Mach. Learn. Res.*, 2023.

Jiangjie Chen, Qianyu He, Siyu Yuan, Aili Chen, Zhicheng Cai, Weinan Dai, Hongli Yu, Qiying Yu, Xuefeng Li, Jiaze Chen, Hao Zhou, and Mingxuan Wang. 2025a. Enigmata: Scaling logical reasoning in large language models with synthetic verifiable puzzles. *CoRR*, abs/2505.19914.

Zhipeng Chen, Yingqian Min, Beichen Zhang, Jie Chen, Jinhao Jiang, Daixuan Cheng, Wayne Xin Zhao, Zheng Liu, Xu Miao, Yang Lu, Lei Fang, Zhongyuan Wang, and Ji-Rong Wen. 2025b. An empirical study on eliciting and improving r1-like reasoning models. *CoRR*, abs/2503.04548.

Zhipeng Chen, Kun Zhou, Xin Zhao, Junchen Wan, Fuzheng Zhang, Di Zhang, and Ji-Rong Wen. 2024. Improving large language models via fine-grained reinforcement learning with minimum editing constraint. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 5694–5711. Association for Computational Linguistics.

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

Stephen Chung, Wenyu Du, and Jie Fu. 2025. Thinker: Learning to think fast and slow. *CoRR*, abs/2505.21097.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, Zhiyuan Liu, Hao Peng, Lei Bai, Wanli Ouyang, Yu Cheng, Bowen Zhou, and Ning Ding. 2025. The entropy mechanism of reinforcement learning for reasoning language models. *CoRR*, abs/2505.22617.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 81 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, and 82 others. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Dong Guo, Faming Wu, Feida Zhu, Fuxing Leng, Guang Shi, Haobin Chen, Haoqi Fan, Jian Wang, Jianyu Jiang, Jiawei Wang, Jingji Chen, Jingjia Huang, Kang Lei, Liping Yuan, Lishu Luo, Pengfei Liu, Qinghao Ye, Rui Qian, Shen Yan, and 81 others. 2025. Seed1.5-vl technical report. *CoRR*, abs/2505.07062.

Mohsen Hariri, Amirhossein Samandar, Michael Hinczewski, and Vipin Chaudhary. 2025. Don't pass@k: A bayesian framework for large language model evaluation. *CoRR*, abs/2510.04265.

Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, Siyuan Li, Liang Zeng, Tianwen Wei, Cheng Cheng, Bo An, Yang Liu, and Yahui Zhou. 2025. Skywork open reasoner 1 technical report. *CoRR*, abs/2505.22312.

Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi,

Junhui Ji, Lihang Pan, and 1 others. 2025. Glm-4.1 v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv preprint arXiv:2507.01006*.

Zhenyu Hou, Xin Lv, Rui Lu, Jiajie Zhang, Yujiang Li, Zijun Yao, Juanzi Li, Jie Tang, and Yuxiao Dong. 2025. Advancing language model reasoning through reinforcement learning and inference scaling. *CoRR*, abs/2501.11651.

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

Jinwu Hu, Zhitian Zhang, Guohao Chen, Xutao Wen, Chao Shuai, Wei Luo, Bin Xiao, Yuanqing Li, and Mingkui Tan. 2025b. Test-time learning for large language models. *CoRR*, abs/2505.20633.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *CoRR*, abs/2310.06825.

Yuhua Jiang, Yuwen Xiong, Yufeng Yuan, Chao Xin, Wenyuan Xu, Yu Yue, Qianchuan Zhao, and Lin Yan. 2025. PAG: multi-turn reinforced LLM self-correction with policy as generative verifier. *CoRR*, abs/2506.10406.

Team Kimi, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, Chuning Tang, Congcong Wang, Dehao Zhang, Enming Yuan, Enzhe Lu, Fengxiang Tang, Flood Sung, Guangda Wei, Guokun Lai, and 75 others. 2025. Kimi k1.5: Scaling reinforcement learning with llms. *CoRR*, abs/2501.12599.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, and 4 others. 2024. Tülu 3: Pushing frontiers in open language model post-training. *CoRR*, abs/2411.15124.

Shilei Li, Meng Li, Jiongming Su, Shaofei Chen, Zhimin Yuan, and Qing Ye. 2021. PP-PG: combining parameter perturbation with policy gradient methods for effective and efficient explorations in deep reinforcement learning. *ACM Trans. Intell. Syst. Technol.*, 12(3):35:1–35:21.

Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. 2025a. Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models. *CoRR*, abs/2505.24864.

Xiaoyuan Liu, Tian Liang, Zhiwei He, Jiahao Xu, Wenxuan Wang, Pinjia He, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2025b. Trust, but verify: A self-verification approach to reinforcement learning with verifiable rewards. *CoRR*, abs/2505.13445.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025c. Understanding r1-zero-like training: A critical perspective. *CoRR*, abs/2503.20783.

Zijun Liu, Peiyi Wang, Runxin Xu, Shirong Ma, Chong Ruan, Peng Li, Yang Liu, and Yu Wu. 2025d. Inference-time scaling for generalist reward modeling. *CoRR*, abs/2504.02495.

Tongxu Luo, Wenyu Du, Jiaxi Bi, Stephen Chung, Zhengyang Tang, Hao Yang, Min Zhang, and Benyou Wang. 2025. Learning from peers in reasoning models. *CoRR*, abs/2505.07787.

Justin K. Miller and Wenjia Tang. 2025. Evaluating LLM metrics through real-world capabilities. *CoRR*, abs/2505.08253.

Yingqian Min, Zhipeng Chen, Jinhao Jiang, Jie Chen, Jia Deng, Yiwen Hu, Yiru Tang, Jiapeng Wang, Xiaoxue Cheng, Huatong Song, Wayne Xin Zhao, Zheng Liu, Zhongyuan Wang, and Ji-Rong Wen. 2024. Imitate, explore, and self-improve: A reproduction report on slow-thinking reasoning systems. *CoRR*, abs/2412.09413.

Kusha Sareen, Morgane M. Moss, Alessandro Sordoni, Rishabh Agarwal, and Arian Hosseini. 2025. Putting the value back in RL: better test-time scaling by unifying LLM reasoners with verifiers. *CoRR*, abs/2505.04842.

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

Amrith Setlur, Matthew Y. R. Yang, Charlie Snell, Jeremy Greer, Ian Wu, Virginia Smith, Max Simchowitz, and Aviral Kumar. 2025. e3: Learning to explore enables extrapolation of test-time compute for llms. *CoRR*, abs/2506.09026.

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

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2025. Hybridflow: A flexible and efficient RLHF framework. In *Proceedings of the Twentieth European Conference on Computer Systems, EuroSys 2025, Rotterdam, The Netherlands, 30 March 2025 - 3 April 2025*, pages 1279–1297. ACM.

Avi Singh, John D. Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J. Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron T. Parisi, Abhishek Kumar, Alexander A. Alemi, Alex Rizkowsky, Azade Nova, Ben Adlam, Bernd Bohnet, Gamaleldin Fathy Elsayed, Hanie Sedghi, and 22 others. 2024. Beyond human data: Scaling self-training for problem-solving with language models. *Trans. Mach. Learn. Res.*, 2024.

Haoxiang Sun, Yingqian Min, Zhipeng Chen, Wayne Xin Zhao, Zheng Liu, Zhongyuan Wang, Lei Fang, and Ji-Rong Wen. 2025. Challenging the boundaries of reasoning: An olympiad-level math benchmark for large language models. *CoRR*, abs/2503.21380.

Yunhao Tang, Kunhao Zheng, Gabriel Synnaeve, and Rémi Munos. 2025. Optimizing language models for inference time objectives using reinforcement learning. *CoRR*, abs/2503.19595.

Luong Quoc Trung, Xinbo Zhang, Zhanming Jie, Peng Sun, Xiaoran Jin, and Hang Li. 2024. Reft: Reasoning with reinforced fine-tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 7601–7614. Association for Computational Linguistics.

Christian Walder and Deep Karkhanis. 2025. Pass@k policy optimization: Solving harder reinforcement learning problems. *CoRR*, abs/2505.15201.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 9426–9439. Association for Computational Linguistics.

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

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8:229–256.

Siye Wu, Jian Xie, Yikai Zhang, Aili Chen, Kai Zhang, Yu Su, and Yanghua Xiao. 2025. ARM: adaptive reasoning model. *CoRR*, abs/2505.20258.

Tian Xie, Zitian Gao, Qingnan Ren, Haoming Luo, Yuqian Hong, Bryan Dai, Joey Zhou, Kai Qiu, Zhirong Wu, and Chong Luo. 2025. Logic-rl: Unleashing LLM reasoning with rule-based reinforcement learning. *CoRR*, abs/2502.14768.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 22 others. 2024. Qwen2.5 technical report. *CoRR*, abs/2412.15115.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, and 16 others. 2025. DAPO: an open-source LLM reinforcement learning system at scale. *CoRR*, abs/2503.14476.

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

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

Kaiwen Zha, Zhengqi Gao, Maohao Shen, Zhang-Wei Hong, Duane S. Boning, and Dina Katabi. 2025. RL tango: Reinforcing generator and verifier together for language reasoning. *CoRR*, abs/2505.15034.

Qiyuan Zhang, Fuyuan Lyu, Zexu Sun, Lei Wang, Weixu Zhang, Zhihan Guo, Yufei Wang, Irwin King, Xue Liu, and Chen Ma. 2025. What, how, where, and how well? A survey on test-time scaling in large language models. *CoRR*, abs/2503.24235.

Yuxiang Zhang, Yuqi Yang, Jiangming Shu, Yuhang Wang, Jinlin Xiao, and Jitao Sang. 2024. Openrft: Adapting reasoning foundation model for domain-specific tasks with reinforcement fine-tuning. *CoRR*, abs/2412.16849.

Xinyu Zhu, Mengzhou Xia, Zhepei Wei, Wei-Lin Chen, Danqi Chen, and Yu Meng. 2025. The surprising effectiveness of negative reinforcement in LLM reasoning. *CoRR*, abs/2506.01347.

Yuxin Zuo, Kaiyan Zhang, Shang Qu, Li Sheng, Xuekai Zhu, Biqing Qi, Youbang Sun, Ganqu Cui, Ning Ding, and Bowen Zhou. 2025. TTRL: test-time reinforcement learning. *CoRR*, abs/2504.16084.

A Detailed Related Work

Reinforcement Learning with Verifiable Rewards. To unleash the potential of LLM reasoning ability, DeepSeek directly employs reinforcement learning with verifiable rewards (RLVR) on DeepSeek-V3, obtaining the large reasoning model DeepSeek-R1-Zero (DeepSeek-AI et al., 2025), which can perform the reasoning process with complex reasoning actions (*e.g.,* reflection and verification). Given the success of the DeepSeek-R1, a surge of studies (Chen et al., 2025b; Zeng et al., 2025; Hu et al., 2025a) have explored the effectiveness of RLVR on the popular open-source LLMs, like Qwen (Yang et al., 2024), Mistral (Jiang et al., 2023), and LLaMA (Dubey et al., 2024). Moreover, the RLVR training paradigm can help LLMs to control their reasoning time (Aggarwal and Welleck, 2025), switch the reasoning pattern (Chung et al., 2025; Wu et al., 2025), enhance the specific performance metric (Tang et al., 2025), and enhance their abilities without supervision (Zuo et al., 2025; Hu et al., 2025b). However, recent work points out that the popular RLVR algorithms (*e.g.,* PPO (Schulman et al., 2017) and GRPO (Shao et al., 2024)) still face serious challenges, like training instability, model collapse, and reward noise (Walder and Karkhanis, 2025; Liu et al., 2025c; Yu et al., 2025; Liu et al., 2025a; Zhu et al., 2025). To mitigate these issues, existing researches propose the optimization on the rollout strategy (Yu et al., 2025), objective function design (Walder and Karkhanis, 2025; Liu et al., 2025c,a), and data selection (Zhu et al., 2025). Specifically, previous work (Walder and Karkhanis, 2025) utilizes Pass@k as the reward on the policy gradient (Williams, 1992) to encourage models to solve hard problems. However, the intrinsic connection between Pass@k RLVR training and LLM exploration ability has not been fully recognized. Thus, we further adopt the Pass@k metric in GRPO and its variants through three approaches (Fig. 9), and derive the analytical solution of advantage values of Pass@k reward in RLVR training. Moreover, according to empirical experiments and theoretical analysis, we discuss the benefits of Pass@k Training in balancing the exploration and exploitation abilities of LLMs during the RLVR training procedure, showing the huge potential of Pass@k RLVR training and pointing out the promising future research directions.

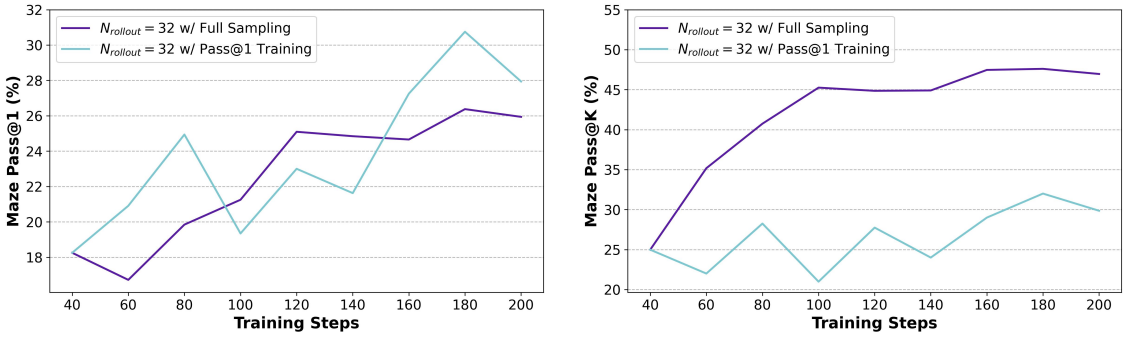
Effective Exploration in Test-time Scaling. Recently, test-time scaling has been proposed to improve the performance of LLMs by consuming more computational resources at inference time (Zhang et al., 2025). Since the LLMs continuously leverages exploration-derived experience to optimize its performance, effective exploration is important and necessary during the test-time scaling process (Hou et al., 2025; Luo et al., 2025). However, existing work reveals that the exploration ability is limited by the corresponding base model, hindering the continuous scaling of model performance (Yue et al., 2025). To mitigate this issue, previous work proposed several approaches, including achieved by adjusting the sampling hyperparameters (Chen et al., 2025b; Hu et al., 2025a; Hou et al., 2025), performing self-verification and self-reflection (Jiang et al., 2025; Liu et al., 2025b; Sareen et al., 2025), or leveraging external models to verify the reasoning process (Liu et al., 2025d; Zha et al., 2025). Beyond these approaches from an external perspective of the model, it is equally important to explore the model's exploration capability through its internal mechanisms. Current studies start from the perspective of the entropy of policy distribution, pointing out that entropy can indicate the exploration ability of LLMs (Cheng et al., 2025; Cui et al., 2025) and high-entropy tokens are vital for model optimization (Wang et al., 2025). Based on these findings, training the critical tokens (Wang et al., 2025) and adding regularization (Hou et al., 2025; Liu et al., 2025a) are employed in the RLVR training process to avoid the degradation of the exploration capability of LLMs. Further, several studies focus on enhancing the exploration abilities of LLMs by selecting useful sampled experience (Zhu et al., 2025; Setlur et al., 2025), integrating entropy into advantage estimation (Cui et al., 2025).

B Experiment Setup

B.1 Details of Downstream Tasks

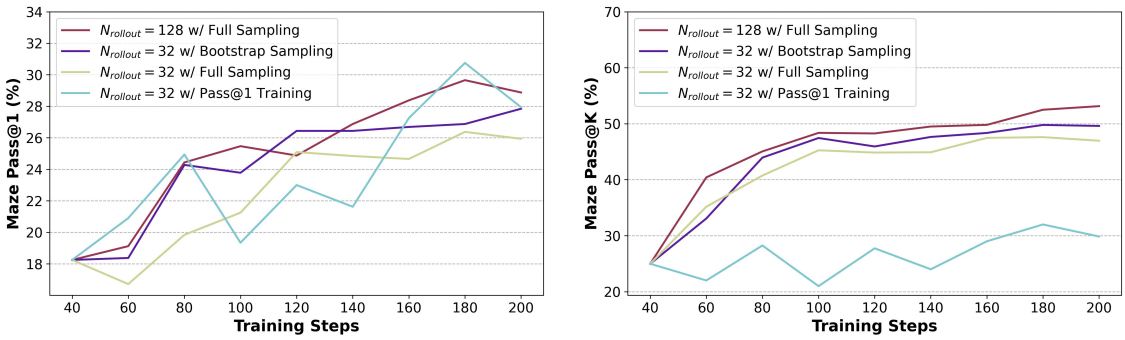
In this section, we present detailed information of each downstream evaluation task.

Maze. We follow the framework proposed by previous work to synthesize the different sizes of mazes. Each maze is represented by text, containing n rows and n columns, a total of $n * n$ characters. Concretely, each of them is one of the following four characters "S", "E", "*", and ".", denoting the start point, destination, available place, and unavailable place, respectively. Given the maze, LLMs can first



(a) Pass@1 Score. (b) Pass@k Score.

Figure 7: Training progress on Maze task of Pass@1 Training and Pass@k Training with Full Sampling.



(a) Pass@1 Scores. (b) Pass@k Scores.

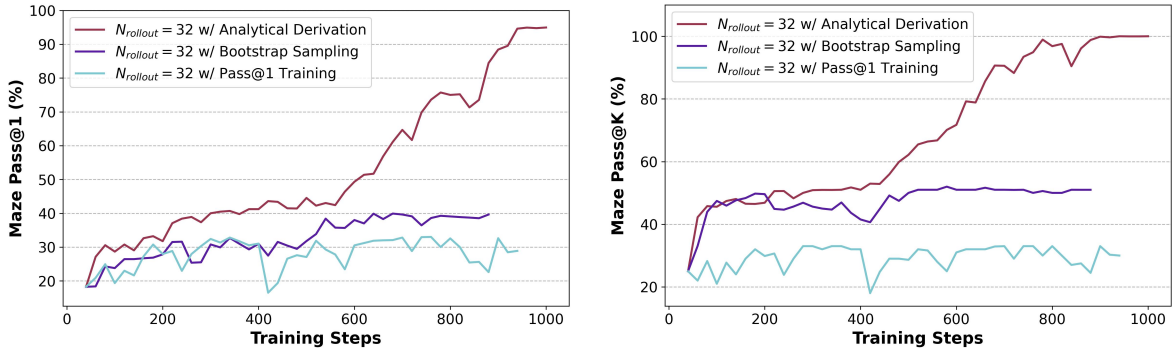
Figure 8: Training progress on Maze task of Pass@1 Training and Pass@k Training with Bootstrap Sampling under various N_{rollout}.

generate the thought or reasoning process and then generate the final answer, which includes one of the four actions "U", "D", "L", and "R", refering to moving up, down, left, and right, respectively. For training data, we construct the mazes with sizes of 9×9, 11×11, 13×13, and 15×15, to increase the diversity of training data. For test data, to evaluate the generalization of the RLVR process, we not only conduct the same sizes of the mazes with the training dataset, but also collect the mazes with sizes of 7×7, 17×17, 19×19, and 21×21. To ensure the validity of the experiment, we performed strict deduplication operations after generating the training and test data. The statistical information of the datasets is presented in Table 4. For better understanding, we present a test instance in Figure 10. To present the empirical insights more clearly, we only showed the results of the 9×9 maze in the above text, and the remaining results are presented in Appendix G.4.

Enigmata. To assess the reasoning and logical abilities of LLMs, Enigmata proposed a comprehensive benchmark that includes the 36 categories of synthetic verifiable puzzles of 7 primary categories, including Crypto Puzzle, Arithmetic Puzzle, Logic Puzzle, Grid Puzzle, Graph Puzzle, Search Puzzle, and Sequential Puzzle. Each category can assess different abilities of LLMs. For better understanding, we present a test instance in Figure 11.

MathVision. MathVision selects 3,040 high-quality problems from human math competitions, each accompanied by relevant images. Solving these problems requires both careful interpretation of the visual information and rigorous mathematical reasoning. MathVision provides a benchmark for assessing a model's multimodal understanding as well as its ability to perform rigorous mathematical reasoning. For better understanding, we present a test instance in Figure 12.

MMMU. MMMU includes college-level reasoning and comprehension tasks across six academic subjects, including Art & Design, Business, Science, Health & Medicine, Humanities & Social Science, and Tech & Engineering. Moreover, MMMU includes a wide range of image types, enabling a



(a) Pass@1 Scores.

(b) Pass@k Scores.

Figure 9: Training progress on Maze task of Pass@1 Training and Pass@k Training with Analytical Derivation and Bootstrap Sampling.

Table 4: The statistical information of the Maze task.

	7×7	9×9	11×11	13×13	15×15	17×17	19×19	21×21
Training Set	-	10,000	10,000	10,000	10,000	-	-	-
Test Set	75	100	100	100	100	100	100	100

comprehensive assessment of a model's capability to process and reason over different forms of visual information. For better understanding, we present a test instance in Figure 13.

B.2 Implementation Details

Training. In our experiment, we adopt $R_{\text{neg}} = 0$ and $R_{\text{pos}} = 1$ for reward function, and employ DAPO (Yu et al., 2025) as the RLVR algorithms. To enhance the efficiency of the training process, we only retain the clip-higher (i.e., $\varepsilon_{\text{low}} = 0.2$ and $\varepsilon_{\text{high}} = 0.28$) and token-level policy gradient loss, and remove other optimizations. For the training hyper-parameters, we set the learning rate for the policy model as 1×10^{-6} with 10 warmup steps, and employ 128, 32, and 32 as prompt batch size BS_{prompt}, mini-batch size BS_{mini}, and rollout times n_{rollout}, respectively. For the reward, the responses that pass the verification (named as positive responses) will be assigned the positive reward $R_{\text{pos}} = 1$, while the other responses (named as negative responses) will be endowed with the negative reward $R_{\text{neg}} = 0$. Additionally, we do not employ any regularization methods, such as KL or Entropy regularization. Besides, for the maximum length of the model response, we set it as 10240 for the puzzle task and 6144 for the mathematical and maze tasks. For other hyper-parameters that are not mentioned above, we adopt the default settings in the verl framework (Sheng et al., 2025). For the training data, we collect the questions that share the same domain as the downstream tasks, *i.e.,* puzzle, maze, and mathematics. For a fair comparison, we train each method until convergence and report the performance of the model at convergence.

For a fair comparison, we train all baselines for 700 steps. For "Pass@k T.+Pass@1 T.", we use 300 steps to Pass@k Training and 400 steps to Pass@1 Training. During 700 steps RLVR, "Pass@k T.+Pass@1 T." outperforms other powerful baselines, including PKPO, ForkingToken and EntropyAdv. During 300 steps RLVR, ForkingToken and EntropyAdv achieves 22.3% and 25.1% on Pass@k scores, respectively. Pass@k Training achieves 29.8% on Pass@k scores, outperforming baselines.

Evaluation. To evaluate the performance of LLMs, we adopt 1.0 and 0.95 as the Temperature and Top_P. For each question, we sample 32 responses from LLMs for the Maze task and sample 8 responses from LLMs for other tasks, and then utilize the sampled responses to compute the Pass@1 and Pass@k scores.

C Discussion About Computation Cost

We employ sampling and analytical derivations to control the additional overhead of Pass@k Training within constant time, without incurring extra computational or temporal cost. Specifically, in our experiments, training the model with Pass@1

Question

You need to solve the following maze.

'*' denotes the wall that you cannot walk through, '.' denotes available area that you can walk through. 'S' denotes the starting point, 'E' denotes the destination.

You need to start from the starting point and cross through the available area to reach the destination. There are four movement actions, including Left, Right, Up, Down.

You need to use L to denote Left movement, R to denote Right movement, U to denote Up movement, and D to denote Down movement.

You can analysis the maze to find the correct path, and you should write the final path in the <answer> </answer>, e.g., <answer> LLRRDUL </answer>.

Maze
```
*********
*.......*
*.*****.*
*E......*
*.*.*.*.*
*.*.*...*
*.*.*.*.*
*S......*
*********
```

Now try to analyze the maze and put the final path in the <answer> </answer>.

Ground Truth

<answer> UUUU <answer>

Figure 10: An example of **Maze** Task.

requires 713.6 GPU hours ("P@1 T."), whereas performing Pass@k Training followed by Pass@1 Training requires 709.3 GPU hours ("P@k T. + P@1 T."). This demonstrates that Pass@k Training does not introduce additional time overhead. We will include the corresponding analysis and discussion in the revised version of the paper.

D Details of Analytical Derivation

We present the details of the analytical derivation procedure mentioned in Sec. 3.4, including the derivation of the average of the group reward, standard deviation of the group reward, and response-relative advantage.

D.1 Derivation of $\hat{A}_{\text{pos}} \geq 0$

Since $\bar{R}^{\text{group}}$ denotes the average value of the reward of each group, it will be less than or equal to the maximum reward of the group, *i.e.,*

$$\bar{R}^{\text{group}} \leq \max(R_{\text{pos}}, R_{\text{neg}}) = R_{\text{pos}} = 1. \tag{10}$$

Based on Eq. 10, we can obtain that $1 - \bar{R}^{\text{group}} \geq 0$. Given that $\sigma^{\text{group}} > 0$, we can derive that,

$$\hat{A}_{\text{pos}} = (1 - \bar{R}^{\text{group}}) \times (\sigma^{\text{group}})^{-1} \geq 0. \tag{11}$$

D.2 Derivation of $\hat{A}_{\text{neg}} \leq 0$

Based on Eq. 6 and Eq. 12, we can derive that,

$$\begin{aligned} \hat{A}_{\text{neg}} &= \left(1 - \bar{R}^{\text{group}} - \frac{\binom{N_{\text{neg}}-1}{k-1}}{\binom{N_{\text{rollout}}-1}{k-1}}\right) \times (\sigma^{\text{group}})^{-1} \\ &= \left(\frac{\binom{N_{\text{neg}}}{k}}{\binom{N_{\text{rollout}}}{k}} - \frac{\binom{N_{\text{neg}}-1}{k-1}}{\binom{N_{\text{rollout}}-1}{k-1}}\right) \times (\sigma^{\text{group}})^{-1}. \end{aligned} \tag{12}$$

In the following, we derive whether

$$\frac{\binom{N_{\text{neg}}}{k}}{\binom{N_{\text{rollout}}}{k}} - \frac{\binom{N_{\text{neg}}-1}{k-1}}{\binom{N_{\text{rollout}}-1}{k-1}} \leq 0, \tag{13}$$

i.e.,

$$\begin{aligned} &\frac{\binom{N_{\text{neg}}}{k}}{\binom{N_{\text{rollout}}}{k}} - \frac{\binom{N_{\text{neg}}-1}{k-1}}{\binom{N_{\text{rollout}}-1}{k-1}} \\ &= \binom{N_{\text{neg}}}{k} \times \binom{N_{\text{rollout}}-1}{k-1} - \binom{N_{\text{neg}}-1}{k-1} \times \binom{N_{\text{rollout}}-1}{k-1} \end{aligned} \tag{14}$$

Since $\binom{N_{\text{rollout}}}{k} \times \binom{N_{\text{rollout}}-1}{k-1} > 0$, we only need to derive $\binom{N_{\text{neg}}}{k} \times \binom{N_{\text{rollout}}-1}{k-1} - \binom{N_{\text{neg}}-1}{k-1} \times \binom{N_{\text{rollout}}-1}{k-1} \leq 0$.

Question

Apply a function to the final input list to generate the output list. Use any preceding inputs and outputs as examples to find what is the function used. All example outputs have been generated using the same function.

Response Format:
- Please output your answer within a code block (```), formatted as a list of numbers, for example: ``` [0, 2, 3] ```

Examples

Example 1:
[6, 4] -> [6, 8, 4]

Example 2:
[8, 3, 2, 0, 9, 7] -> [8, 5, 3, 2, 0, 9, 7]

Example 3:
[1, 2, 6, 0, 9, 3] -> [1, 5, 2, 6, 0, 9, 3]

Example 4: [9, 7, 8] -> [9, 8, 7, 8]

Example 5:
[1, 9, 6, 5, 0, 3, 8, 4, 7, 2] -> [1, 5, 9, 6, 5, 0, 3, 8, 4, 7, 2]

Example 6:
[9, 8] -> [9, 8, 8]

Test Problem:
[7, 4, 6, 8, 0, 1, 3] ->

Ground Truth

```[7, 5, 4, 6, 8, 0, 1, 3]```

Figure 11: An example of **Enigmata** Task.

$$
\begin{aligned}
&\binom{N_{\text{neg}}}{k} \times \binom{N_{\text{rollout}}-1}{k-1} - \binom{N_{\text{neg}}-1}{k-1} \times \binom{N_{\text{rollout}}-1}{k-1} \\
=&\frac{N_{\text{neg}}! \times (N_{\text{rollout}}-1)!}{k!(N_{\text{neg}}-k)! \times (k-1)!(N_{\text{rollout}}-k)!} \\
&- \frac{(N_{\text{neg}}-1)! \times N_{\text{rollout}}!}{(k-1)!(N_{\text{neg}}-k)! \times k!(N_{\text{rollout}}-k)!} \\
=&\frac{(N_{\text{neg}}-1)!(N_{\text{rollout}}-1)!}{k!(k-1)!(N_{\text{neg}}-k)!(N_{\text{rollout}}-k)!} \times (N_{\text{neg}} - N_{\text{rollout}}).
\end{aligned} \tag{15}
$$

Since

$$
\frac{(N_{\text{neg}}-1)!(N_{\text{rollout}}-1)!}{k!(k-1)!(N_{\text{neg}}-k)!(N_{\text{rollout}}-k)!} > 0, \tag{16}
$$

and

$$
N_{\text{neg}} \leq N_{\text{rollout}}, \tag{17}
$$

we have $\binom{N_{\text{neg}}}{k} \times \binom{N_{\text{rollout}}-1}{k-1} - \binom{N_{\text{neg}}-1}{k-1} \times \binom{N_{\text{rollout}}-1}{k-1} \leq 0$. Therefore, we can derive that

$$
\hat{A}_{\text{neg}} \leq 0. \tag{18}
$$

D.3 Average of Group Reward

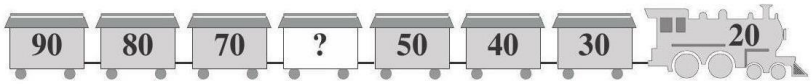
$$
\begin{aligned}
\bar{R}^{\text{group}} &= \frac{1}{N_{\text{total}}^{\text{group}}} \times (N_{\text{pos}}^{\text{group}} \times R_{\text{pos}} + N_{\text{neg}}^{\text{group}} \times R_{\text{neg}}) \\
&= \frac{1}{\binom{N_{\text{rollout}}}{K}} \times \left(\left(\binom{N_{\text{rollout}}}{K} - \binom{N_{\text{neg}}}{K}\right) \times 1 + \left(\binom{N_{\text{neg}}}{K}\right) \times 0\right) \\
&= 1 - \frac{\binom{N_{\text{neg}}}{K}}{\binom{N_{\text{rollout}}}{K}}.
\end{aligned} \tag{19}
$$

D.4 Standard Deviation of Group Reward

$$
\begin{aligned}
\sigma^{\text{group}} =& \Big(\frac{1}{N_{\text{rollout}}} \times (N_{\text{pos}}^{\text{group}} \times (\bar{R}^{\text{group}} - R_{\text{pos}})^2 \\
&+ N_{\text{neg}}^{\text{group}} \times (\bar{R}^{\text{group}} - R_{\text{neg}})^2)\Big)^{\frac{1}{2}} \\
=& \Big(\left(\binom{N_{\text{rollout}}}{K} - \binom{N_{\text{neg}}}{K}\right) \times \Big(2 - \frac{\binom{N_{\text{neg}}}{K}}{\binom{N_{\text{bootstrap}}}{K}}\Big)^2 \div \binom{N_{\text{rollout}}}{K} \\
&+ \binom{N_{\text{neg}}}{K} \times \left(1 - \frac{\binom{N_{\text{neg}}}{K}}{\binom{N_{\text{rollout}}}{K}}\right)^2 \div \binom{N_{\text{rollout}}}{K}\Big)^{\frac{1}{2}} \\
=& \Big(\Big(1 - \frac{\binom{N_{\text{neg}}}{K}}{\binom{N_{\text{bootstrap}}}{K}}\Big)\Big(\frac{\binom{N_{\text{neg}}}{K}}{\binom{N_{\text{rollout}}}{K}}\Big)^2 + \frac{\binom{N_{\text{neg}}}{K}}{\binom{N_{\text{rollout}}}{K}}\Big(1 - \frac{\binom{N_{\text{neg}}}{K}}{\binom{N_{\text{rollout}}}{K}}\Big)^2\Big)^{\frac{1}{2}} \\
=& \Big(\Big(1 - \frac{\binom{N_{\text{neg}}}{K}}{\binom{N_{\text{rollout}}}{K}}\Big)\Big(\frac{\binom{N_{\text{neg}}}{K}}{\binom{N_{\text{rollout}}}{K}}\Big)\Big(1 - \frac{\binom{N_{\text{neg}}}{K}}{\binom{N_{\text{rollout}}}{K}} + \frac{\binom{N_{\text{neg}}}{K}}{\binom{N_{\text{rollout}}}{K}}\Big)\Big)^{\frac{1}{2}} \\
=& \Big(\Big(1 - \frac{\binom{N_{\text{neg}}}{K}}{\binom{N_{\text{rollout}}}{K}}\Big)\Big(\frac{\binom{N_{\text{neg}}}{K}}{\binom{N_{\text{rollout}}}{K}}\Big)\Big)^{\frac{1}{2}} = (\bar{R}^{\text{group}}(1 - \bar{R}^{\text{group}}))^{\frac{1}{2}}.
\end{aligned}
$$
```

Question

Which number should be written in place of the question mark?



Ground Truth

60

Figure 12: An example of **MathVision** Task.

Question

Each of the following situations relates to a different company.

		Company A	Company B	Company C	Company D
1	Revenues	?	$1,480,500	$103,950	$1,054,116
2	Expenses	$455,490	1,518,300	78,120	?
3	Gains	0	?	4,725	8,505
4	Losses	32,760	0	5,670	39,312
5	Net Income or (Loss)	32,130	39,690	?	(58,275)

For company B, find the missing amounts.
(A) $63,020
(B) $58,410
(C) $71,320
(D) $77,490

Ground Truth

D

Figure 13: An example of **MMMU** Tasks.

D.5 Response-Relative Advantage

$$
\begin{aligned}
\hat{A}_{\text{pos}} =& \frac{1}{\binom{N_{\text{rollout}}-1}{K-1}} \times \left(\binom{N_{\text{rollout}}-1}{K-1} \times \hat{A}_{\text{pos}}^{\text{group}} + 0 \times \hat{A}_{\text{neg}}^{\text{group}}\right) \\
=& \frac{1-\bar{R}^{\text{group}}}{\sigma^{\text{group}}}. \\
\hat{A}_{\text{neg}} =& \frac{1}{\binom{N_{\text{rollout}}-1}{K-1}} \times \left(\left(\binom{N_{\text{rollout}}-1}{K-1} - \binom{N_{\text{neg}}-1}{K-1}\right) \times \hat{A}_{\text{pos}}^{\text{group}}\right. \\
&\left. + \binom{N_{\text{neg}}-1}{K-1} \times \hat{A}_{neg}^{\text{group}}\right) \\
=& \left(\left(1 - \frac{\binom{N_{\text{neg}}-1}{K-1}}{\binom{N_{\text{rollout}}-1}{K-1}}\right) \times \hat{A}_{\text{pos}}^{\text{group}} + \frac{\binom{N_{\text{neg}}-1}{K-1}}{\binom{N_{\text{rollout}}-1}{K-1}} \times \hat{A}_{\text{neg}}^{\text{group}}\right) \\
=& \left(\left(1 - \frac{\binom{N_{\text{neg}}-1}{K-1}}{\binom{N_{\text{rollout}}-1}{K-1}}\right) \times \frac{1-\bar{R}^{\text{group}}}{\sigma^{\text{group}}}\right. \\
&\left. + \frac{\binom{N_{\text{neg}}-1}{K-1}}{\binom{N_{\text{rollout}}-1}{K-1}} \times \left(-\frac{\bar{R}^{\text{group}}}{\sigma^{\text{group}}}\right)\right) \\
=& \left(1 - \bar{R}^{\text{group}} - \frac{\binom{N_{\text{neg}}-1}{K-1}}{\binom{N_{\text{rollout}}-1}{K-1}}\right) \times (\sigma^{\text{group}})^{-1}.
\end{aligned}
\tag{20}
$$

D.6 Comparison of Bias and Variance between Analytical Derivation and Full Sampling

In this section, we provide a simple theoretical analysis of the algorithm's bias and variance.

For model π_θ, we roll out n_{rollout} times, generating n_{pos} positive responses and n_{neg} negative responses. We assume that the distribution obtained from sampling is $\pi_{\theta'}$. In the training process of RLVR, we use the $\mathbb{E}[\pi_{\theta'}]$ to estimate $\mathbb{E}[\pi_\theta]$. Since $\pi_{\theta'}$ is obtained by sampling from π_θ, there will be a bias in the estimation process, *i.e.,*

$$\mathbb{E}[\pi_\theta] = \mathbb{E}[\pi_{\theta'}] + b_1, \tag{21}$$

where $b_1 \neq 0$ and refers to the bias. In Full Sampling, $\pi_{\theta'}$ is estimated using a one-step sampling approach, which will also bring the bias, *i.e.,*

$$\mathbb{E}_{\text{FS}}[\pi_{\theta'}] = \mathbb{E}[\pi_{\theta'}] + b_2, \tag{22}$$

where $b_2 \neq 0$ and refers to the bias from the Full Sampling process. In contrast, the Analytical Derivation directly computes the analytical solution of $\pi_{\theta'}$ and constitutes an unbiased estimator of $\pi_{\theta'}$. Therefore, we can conclude that

$$\mathbb{E}_{\mathrm{AD}}[\pi_{\theta'}] = \mathbb{E}[\pi_{\theta'}]. \tag{23}$$

Since the Analytical Derivation introduces no additional bias, its estimate of the original distribution π_θ exhibits lower bias. Meanwhile, the sampling operation in Full Sampling introduces additional variance, resulting in Full Sampling having higher variance than Analytical Derivation.

E Pseudo Code for Pass@k Training

We present the pseudo code for Pass@k Training with full sampling (Algorithm 1), bootstrap sampling (Algorithm 2), and analytical derivation (Algorithm 3).

F Curves of Advantage Function

We present the curves of the advantage function of different training approaches in Figure 14, including Pass@k Training w/o easy problems, Pass@k Training w/ combination, exceeding Pass@k Training, and combination training.

G Experiments on Various LLMs and Tasks

In this section, to further verify the effectiveness of Pass@k Training, we first present the significance test for the results in Sec. 4.4, and provide the performance of various LLMs trained through Pass@k Training on Mathematical Tasks (*i.e.,* AIME 2024 (AIME2024, 2024), AIME 2025 (AIME2025, 2025), and OlymMATH (Sun et al., 2025)) and Synthetic Puzzle Task (*i.e.,* Enigmata (Chen et al., 2025a)).

G.1 Significance Test

To strengthen the paper's credibility, we provided the mean and standard deviation of multiple runs for different algorithms on the Enigmata task in Table 5.

By comparing the performance of "P@1 T." and "P@k T." on the Pass@k score, we find that the difference in their means is more than three times the standard deviation, indicating with 99.7% confidence that Pass@k training achieves higher Pass@k scores than Pass@1 training. Similarly, by comparing "+P@1 T." and "+P@k T. + P@1 T." on the Pass@1 score, we find that the difference in their means exceeds twice the standard deviation, providing 95.4% confidence that our method achieves better Pass@1 scores than the traditional Pass@1 training approach.

Similarly, by comparing the performance of "GPT-4o-1120" and "+P@k T. + P@1 T.", we find that the difference in their average Pass@1 score also exceeds twice the standard deviation, demonstrating that our method enables the 7B model to outperform larger models.

G.2 Pass@k Training on Mathematical Tasks

We follow the experiment settings described in Appendix B.2 to perform Pass@k Training on LLaMA models (Dubey et al., 2024) (*i.e.,* LLaMA3.2-3B-Instruct and LLaMA3.1-8B-Instruct) and DeepSeek-R1-Distill-Qwen (DeepSeek-AI et al., 2025) (*i.e.,* 1.5B and 7B version). For LLaMA models, we set the maximum prompt length and response length as 2048 and 6144, respectively. For DeepSeek-R1-Distill-Qwen, we extend the response length to 10240. Specifically, to adapt the LLMs to the mathematical tasks, we adopt the training data used in previous work (Chen et al., 2025b) during the RLVR training process. Besides, we follow the settings in Appendix B.2 to perform the evaluation, and present the results in Table 6. Since the single turn of Pass@k Training followed by Pass@1 Training can significantly improve the Pass@1 performance of LLMs, we conduct the experiment about multiple turns of the above training process in Table 6, named as "(P@k T. + P@1 T.) $\times$ 2".

G.3 Pass@k Training on Enigmata Task

We follow the experiment settings described in Appendix B.2 to perform Pass@k Training on various LLMs (*i.e.,* LLaMA3.2-3B-Instruct (Dubey et al., 2024) and LLaMA3.1-8B-Instruct (Dubey et al., 2024)), and set the maximum of the prompt length and response length as 4096 and 4096, respectively. The results are presented in Table 7. For evaluation, we follow the settings described in Appendix B.2.

G.4 Pass@k Training on Maze Task

In this part, we present the full results of Pass@k Training on the Maze task in Table 8. Without any RLVR training, it is really difficult for the model to solve the Maze task. Thus, we do not report the performance of the backbone model.

Algorithm 1: The Pseudo Code for Pass@k Training with Full Sampling.

Input : A tensor of reward $\mathcal{R} \in \mathbb{R}^{N_{\text{rollout}}}$ of the responses for the problem, the number of the rollouted responses N_{rollout}, and the k for Pass@k metric.
Output : A tensor of estimated advantages of the responses for this problem $\hat{\mathcal{A}} \in \mathbb{R}^{N_{\text{rollout}}}$.

```
# Construct the groups and discard the redundant instances.
```
Separate $\mathcal{R} \in \mathbb{R}^{N_{\text{rollout}}}$ into $\lfloor \frac{N_{\text{rollout}}}{K} \rfloor$ group and each group contains k instances.
Compute the reward of the groups $\mathcal{R}^{\text{group}} \in \mathbb{R}^{\lfloor \frac{N_{\text{rollout}}}{K} \rfloor}$ using Eq. 3.

```
# Follow GRPO advantage estimation method to compute group-relative advantage.
```
Compute the average reward of the groups $\bar{R}^{\text{group}}$ using Eq. 1.
Compute the standard deviation of the groups σ^{group} using Eq. 1.
Based on $\bar{R}^{\text{group}}$ and σ^{group}, compute the group-relative advantage $\hat{\mathcal{A}}^{\text{group}}$ using Eq. 1.

```
# Compute response-relative advantage.
```
Assign the $\hat{\mathcal{A}}^{\text{group}}$ to the responses that the group contains, obtaining the response-relative advantage $\hat{\mathcal{A}}$.

Algorithm 2: The Pseudo Code for Pass@k Training with Bootstrap Sampling.

Input : A tensor of reward $\mathcal{R} \in \mathbb{R}^{N_{\text{rollout}}}$ of the responses for the problem, the number of the rollouted responses N_{rollout}, and the k for Pass@k metric.
Output : A tensor of estimated advantages of the responses for this problem $\hat{\mathcal{A}} \in \mathbb{R}^{N_{\text{rollout}}}$.

```
# Construct the groups through bootstrap sampling.
```
for *i from* 1 *to* N^{group} **do**
  Randomly sample k instances from $\mathcal{R}$ to construct the i-th group.
  Compute the reward of i-th group using Eq. 3.
Obtain the reward of the groups $\mathcal{R}^{\text{group}} \in \mathbb{R}^{N^{\text{group}}}$.

```
# Follow GRPO advantage estimation method to compute group-relative advantage.
```
Compute the average reward of the groups $\bar{R}^{\text{group}}$ using Eq. 1.
Compute the standard deviation of the groups σ^{group} using Eq. 1.
Based on $\bar{R}^{\text{group}}$ and σ^{group}, compute the group-relative advantage $\hat{\mathcal{A}}^{\text{group}}$ using Eq. 1.

```
# Calculate response-relative advantage.
```
Based on $\hat{\mathcal{A}}^{\text{group}}$, compute response-relative advantage $\hat{\mathcal{A}}$ using Eq. 4.

G.5 Pass@k Training on MoE model on Multi-modal Reasoning Tasks

To further evaluate the effectiveness of Pass@k Training, we conducted tests on an in-house model. Unlike the language dense model used in the previous evaluation, this model is a multi-modal MoE model. Table 9 presents the model's test results on MathVision and MMMU. According to the evaluation results, we can observe that the Pass@k performance of the model can be significantly improved, *i.e.,* from 76.4 to 80.0, showing the improvement in the model's exploration ability. After Pass@k Training, the following Pass@1 Training further enhances the Pass@1 scores of the model. These results show the effectiveness of Pass@k Training on different model architectures and different type of downstream tasks.

H Difference Between Pass@1 and Pass@k Training

H.1 Analysis Based on Advantage Value Curves

To analyze why Pass@k Training can help LLMs escape the local optimum, we first visualize the advantage curves of Pass@1 Training and Pass@k Training across responses with different correctness levels, as in GRPO and its variants, the advantage value depends solely on the correctness of the model's response. Furthermore, we observe that during the optimization process, the advantage value is directly multiplied by the gradient and can be interpreted as a scaling factor for the gradient. In this context, a larger absolute value of the advantage indicates a greater scaling of the gradient, and thus a larger update step for the corresponding sample. This implies that the model places greater optimization effort on samples with higher advantage magnitudes. Therefore, we argue that

Algorithm 3: The Pseudo Code for Pass@k Training with Analytical Derivation.

Input : A tensor of reward $\mathcal{R} \in \mathbb{R}^{N_{\text{rollout}}}$ of the responses for the problem, the number of the rollouted responses N_{rollout}, and the k for Pass@k metric.
Output : A tensor of estimated advantages of the responses for this problem $\hat{\mathcal{A}} \in \mathbb{R}^{N_{\text{rollout}}}$.

```
# Calculate the average and standard deviation of the group reward scores.
```
Compute the average reward of the groups $\bar{R}^{\text{group}}$ using Eq. 6.
Compute the standard deviation of the groups σ^{group} using Eq. 6.

```
# Calculate response-relative advantage.
```
Compute the advantage of the positive responses $\hat{A}_{\text{pos}}$ using Eq. 8.
Compute the advantage of the negative responses $\hat{A}_{\text{neg}}$ using Eq. 12.
Based on $\hat{A}_{\text{pos}}$, $\hat{A}_{\text{neg}}$, and $\mathcal{R}$, assign the advantage to each instance, obtaining response-relative advantage $\hat{\mathcal{A}}$.

Table 5: The mean and standard deviation of multiple runs for different algorithms on the Enigmata task.

Methods	Pass@1 Score		Pass@k Score	
	mean@8	std@8	mean@8	std@8
GPT-4o-1120	14.2	-	-	-
Qwen2.5-7B-Instruct + P@1 T.	12.9	1.91	21.3	0.36
Qwen2.5-7B-Instruct + P@k T.	17.9	4.70	29.8	0.73
Qwen2.5-7B-Instruct + P@k T. + P@1 T.	30.8	7.44	40.6	0.28

the absolute value of the advantage is also an important aspect worthy of investigation. Based on this insight, and to simplify the analysis, we compute the sum of the absolute advantage values across all responses, as defined below:

$$\eta = N_{\text{pos}} \times |\hat{A}_{\text{pos}}| + N_{\text{neg}} \times |\hat{A}_{\text{neg}}|, \qquad (24)$$

The curves of η (named as *Sum of Absolute Advantage*) are added to our visualization and presented in Fig. 14a and Fig. 14b. Comparing the curves of η of Pass@1 Training and Pass@k Training, we can observe that there are three major differences.

Maximum of Sum of Absolute Advantage η. The maximum of η of the Pass@1 Training approach is much higher than ones of Pass@k Training approach. As we discussed in Sec. 4.3, the maximum advantage values might affect training efficiency, and adding the coefficient on the loss function to adjust the advantage values can mitigate this issue. Thus, the maximum is not the critical factor that helps Pass@k Training outperform Pass@1 Training.

Argmax of Sum of Absolute Advantage η. According to the curves in Fig. 14a and Fig. 14b, the argmax of η are significantly different between Pass@1 and Pass@8 Training. For Pass@1 Training, the maximum of η appears at the position of 50% accuracy (*i.e.,* $N_{\text{pos}} = 0.5 \times N_{\text{rollout}}$), while the position of maximum of η is 25% accuracy (*i.e.,* $N_{\text{pos}} = 0.25 \times N_{\text{rollout}}$). This phenomenon suggests that Pass@k Training focuses on optimizing harder problems, while Pass@1 Training focuses on problems with medium difficulty. This further demonstrates that Pass@k Training tends to guide the model toward solving previously unsolved or difficult problems, rather than overfitting to those it has already mastered.

Trend of Sum of Absolute Advantage η. Another key difference between the function curves of Pass@1 and Pass@k Training lies in the trend of the function itself. In the η curve of Pass@k Training, the value increases until it reaches a peak, and then gradually decreases to zero. Under this setting, when the problem is relatively easy (*i.e.,* the correctness is higher than 60%), the optimization strength applied by the model (as indicated by the value of η) becomes smaller than that for harder problems. This further demonstrates that Pass@k Training focuses more on optimizing problems the model has not yet mastered. In contrast, during Pass@1 Training, the η curve is symmetric around the point of maximum value, indicating that the training process allocates equal attention to both easy and hard problems.

H.2 Analysis Based on Model Performance

As we discussed in the previous section, the effectiveness of the argmax and the trend of the sum of absolute advantage η still remain unclear. Thus,

Table 6: Pass@1/Pass@k Performance on mathematical tasks of LLaMA and DeepSeek-R1-Distill-Qwen models trained through different RLVR approaches. "P@1 T." and "P@k T." denote the Pass@1 Training and Pass@k Training with analytical derivation, respectively. "(P@k T. + P@1 T.) × 2" refers to that the process of Pass@k Training followed by Pass@1 Training is repeated twice.

	AIME 2024	AIME 2025	OlymMATH-Easy	OlymMATH-Hard	Avg.
RLVR on LLaMA3.2-3B-Instruct (Pass@1/Pass@k)					
Baseline	1.5/17.3	0.1/2.1	1.7/14.4	1.1/9.2	1.1/10.8
+ P@1 T.	13.6/26.7	1.1/6.6	3.8/4.0	2.0/6.3	5.1/10.9
+ P@k T.	12.7/32.0	1.7/12.9	3.7/8.8	1.7/7.7	5.0/**15.4**
+ P@k T. + P@1 T.	14.6/32.1	1.3/8.6	4.1/7.7	2.0/7.5	**5.5**/14.0
RLVR on LLaMA3.1-8B-Instruct (Pass@1/Pass@k)					
Baseline	3.4/17.9	0.2/4.3	0.8/7.5	0.5/7.6	1.0/9.3
+ P@1 T.	4.4/32.1	0.9/7.7	1.4/4.1	1.1/6.2	2.0/12.5
+ P@k T.	7.1/40.0	1.8/10.6	1.5/8.9	1.4/8.2	3.0/**17.0**
+ P@k T. + P@1 T.	8.7/29.7	0.9/8.7	1.8/7.9	1.6/6.8	**3.3**/13.3
RLVR on DeepSeek-R1-Distill-Qwen-1.5B (Pass@1/Pass@k)					
Baseline	22.7/61.4	20.5/37.2	6.6/36.7	0.6/5.2	12.6/35.1
+ P@1 T.	36.7/76.0	28.8/49.4	16.7/51.7	2.5/17.5	21.2/48.7
+ P@k T.	36.5/79.3	27.0/55.5	17.6/59.3	2.4/17.4	20.9/52.9
+ P@k T. + P@1 T.	42.3/71.7	30.4/57.8	20.7/60.5	3.4/18.9	24.2/52.2
+ (P@k T. + P@1 T.) × 2	44.2/77.2	31.5/57.7	22.6/62.7	4.4/21.2	**25.7/54.7**
RLVR on DeepSeek-R1-Distill-Qwen-7B (Pass@1/Pass@k)					
Baseline	43.2/80.9	31.5/59.1	22.2/66.0	1.4/13.7	24.6/54.9
+ P@1 T.	48.5/79.5	35.5/59.4	27.9/69.1	3.1/22.5	28.8/57.6
+ P@k T.	48.2/80.9	36.5/66.7	28.1/72.7	3.3/23.3	29.0/**60.9**
+ P@k T. + P@1 T.	50.3/81.0	39.3/61.9	32.3/68.9	3.5/22.7	**31.4**/58.6

Table 7: Enigmata Pass@1/Pass@k Performance of LLaMA models trained on different RLVR approaches. "P@1 T." and "P@k T." denote the Pass@1 Training and Pass@k Training with analytical derivation, respectively.

	Crypto	Arithmetic	Logic	Grid	Graph	Search	Sequential	Overall
RLVR on LLaMA3.2-3B-Instruct (Pass@1/Pass@k)								
Baseline	0.0/0.0	0.2/1.6	19.3/44.3	2.7/4.7	1.7/8.8	5.4/11.1	0.4/1.8	3.1/7.3
+ P@1 T.	0.0/0.0	0.2/1.3	19.7/27.0	17.4/18.0	5.3/12.8	12.9/14.5	9.8/10.7	11.1/13.0
+ P@k T.	0.0/0.0	0.2/0.7	22.0/31.0	17.3/18.1	6.1/14.2	12.2/16.6	10.7/12.0	11.5/**14.1**
+ P@k T. + P@1 T.	0.0/0.0	0.4/2.2	22.8/27.7	16.7/17.4	6.5/13.5	14.6/16.0	12.0/13.0	**12.3**/14.0
RLVR on LLaMA3.1-8B-Instruct (Pass@1/Pass@k)								
Baseline	0.0/0.0	0.1/1.1	21.6/41.7	3.7/4.6	1.5/7.8	6.0/17.0	1.2/5.0	3.8/9.0
+ P@1 T.	0.0/0.0	0.1/0.9	29.2/38.0	12.1/13.2	3.8/8.5	12.1/14.7	5.3/7.4	8.7/11.1
+ P@k T.	0.0/0.0	0.1/0.9	30.5/39.3	12.9/14.8	5.5/12.2	12.2/14.7	7.5/10.9	9.9/13.0
+ P@k T. + P@1 T.	0.0/0.0	0.2/1.1	34.4/44.7	12.5/14.2	7.5/17.8	13.2/15.7	8.7/10.3	**10.8/13.7**

in this section, we design the corresponding experiments to analyze their effectiveness, based on model performance. Additionally, we designed two training methods that serve as intermediates between Pass@1 and Pass@k Training, *i.e.,* removing the advantage values of the easy problems and combining the advantage estimation approaches of Pass@1 and Pass@k based on the accuracy of the current prompt. The curves of $\hat{A}_{\text{pos}}$, $\hat{A}_{\text{neg}}$, and η of these four training approaches are presented in Fig. 14c and Fig. 14d.

First, when the correctness of a response is high, we design the advantage function to decrease gradually toward zero. This setting allows the training reward to increase steadily during the optimization process, indicating that the model avoids getting stuck in a local optimum (*i.e.,* the blue line and purple line). When this optimization is removed, the reward on the training set fails to continue increasing, suggesting that the model has already converged to a local optimum and is no longer learning new knowledge during the RLVR process (*i.e.,* the red line and green line). This phenomenon suggests that excessive learning from easy examples is a key factor causing the model to fall into local optima. Therefore, reducing the degree of learning from easy questions can help prevent the model from getting trapped in such suboptimal solutions.

Second, simply setting the reward for easy questions to zero is not sufficient to effectively prevent

Table 8: The Pass@1/Pass@k performance of Qwen2.5-7b-Instruct trained on different approaches on various Maze sizes. "P@1 T." and "P@k T." denote the Pass@1 Training and Pass@k Training with analytical derivation, respectively. "FS", "BS", and "AD" denote the full sampling, bootstrap sampling, and analytical derivation, respectively. We report the best performance during the training process of each approach.

	7×7	9×9	11×11	13×13	15×15	17×17	19×19	21×21	Avg.
+ P@1 T.	36.0/38.2	32.4/33.0	10.6/11.0	14.0/14.0	8.1/9.0	5.0/5.0	2.0/2.0	3.0/3.0	13.9/14.4
+ P@k T. w/ FS	34.6/67.4	26.4/47.6	13.7/26.0	11.0/18.5	8.6/17.6	3.0/7.6	2.2/7.9	1.9/5.6	12.7/24.7
+ P@k T. w/ BS	45.3/70.6	37.8/51.0	15.4/27.0	12.8/20.7	12.3/19.9	3.3/8.8	4.8/9.0	2.3/6.3	16.8/26.7
+ P@k T. w/ AD	86.8/98.2	94.6/100.0	75.2/98.3	55.2/84.6	39.2/72.0	10.5/29.2	16.7/28.3	3.5/9.7	**47.7/65.0**

Table 9: Pass@1/Pass@k Performance of an in-house MoE model trained on different RLVR approaches.

Pass@1/Pass@k	**MathVision**	**MMMU**	**Avg.**
baseline	54.6/72.5	71.2/80.2	62.9/76.4
+ Pass@1 Training	55.3/74.0	72.0/83.7	63.7/78.9
+ Pass@k Training	53.9/75.6	72.0/84.3	63.0/80.0
+ Pass@k Training + Pass@1 Training	56.4/76.8	72.3/84.5	**64.4/80.7**

the model from over-optimizing on them; it merely delays the point at which the model falls into a local optimum. As shown in Fig. 15, removing the optimization for easy questions (represented by the red line) leads to higher training rewards and better test performance compared to the baseline (represented by the green line). However, both curves exhibit similar trends: after an initial phase of improvement, model performance gradually plateaus, making further progress difficult.

Third, regarding the choice of the argmax position of the η function, a comparison of the curves in Fig. 15 reveals that shifting the argmax forward leads to higher optimization efficiency. Specifically, the model is able to escape from local optima more quickly, and a turning point in training reward appears earlier. This phenomenon suggests that hard problems contribute more significantly to model improvement and yield better optimization effects. Assigning greater optimization strength to harder problems can thus effectively enhance training efficiency, allowing the model to achieve better performance with fewer training steps.

Based on the above results and discussions, several preliminary conclusions can be made, *i.e.,* the argmax of ηinfluences the training efficiency and the trend of η prevents model from falling into local optimum. Besides, it is important to note that this is only our initial conclusion. More comprehensive experiments tailored to specific tasks and scenarios are required for further validation.

I Further Analysis about Pass@k Training

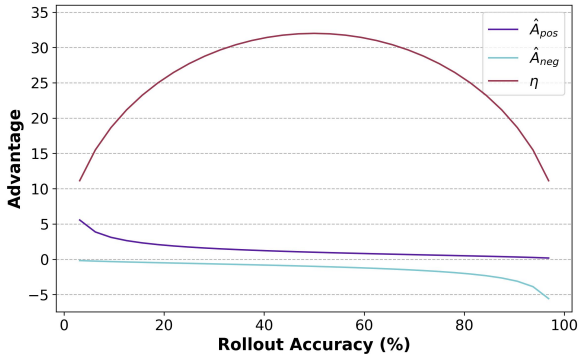
I.1 Can Pass@k Training be Integrated into Different RLVR Algorithms?

A key open question is whether Pass@k Training can be integrated into other reinforcement learning algorithms. This determines its practicality for real-world deployment and its ability to generalize across diverse application scenarios. To assess the adaptability of Pass@k Training, we combine it with multiple popular RLVR algorithms and test whether Pass@k Training can enhance the model's exploration ability when built upon different backbone algorithms, including DAPO (Yu et al., 2025), GRPO (Shao et al., 2024), and ForkingToken (Wang et al., 2025). According to the experiment results in Table 10, we can observe that Pass@k Training can be adapted to different RLVR algorithms, achieving an improvement in the Pass@k score of LLM. This phenomenon indicates that current popular RLVR algorithms can integrate the Pass@k Training to further improve the exploration ability of LLMs.

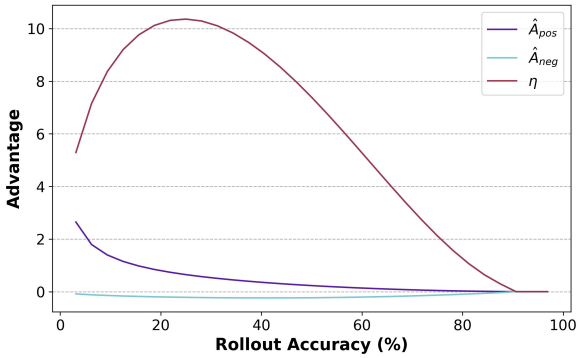
Takeaway from Appendix I.1. Pass@k Training can be flexibly integrated into various RLVR algorithms, consistently boosting LLMs' Pass@k performance and enhancing their exploration ability.

I.2 What is the Generalization Ability of LLMs After Pass@k Training?

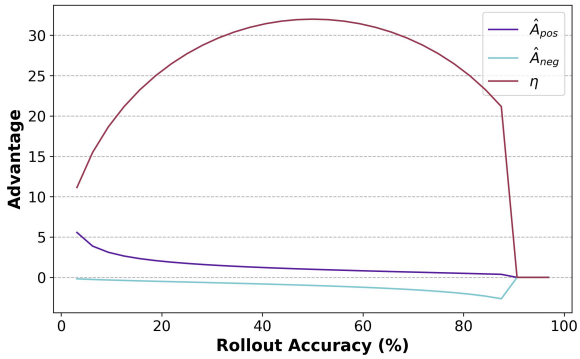
To analyze the generalization ability of Pass@k Training, we conduct the corresponding experi-



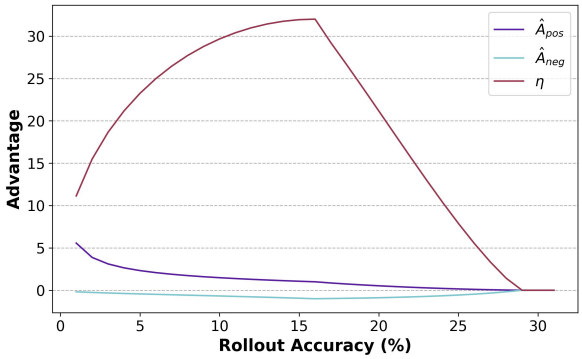
(a) Pass@1 Training.



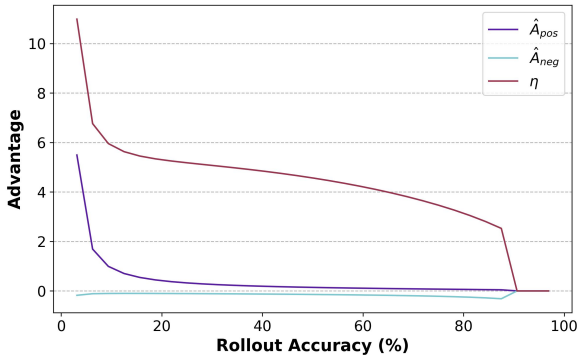
(b) Pass@k Training.



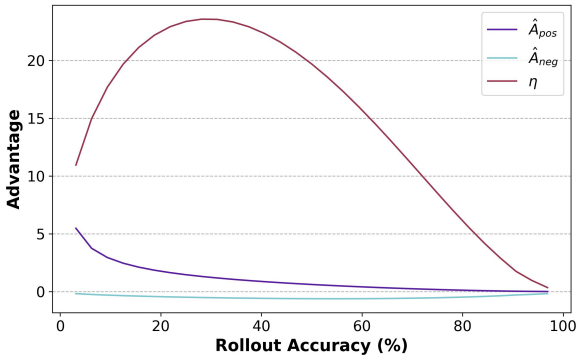
(c) Pass@k Training w/o Easy Problems.



(d) Pass@k Training w/ Combination.



(e) Exceeding Pass@k Training.



(f) Combination Training.

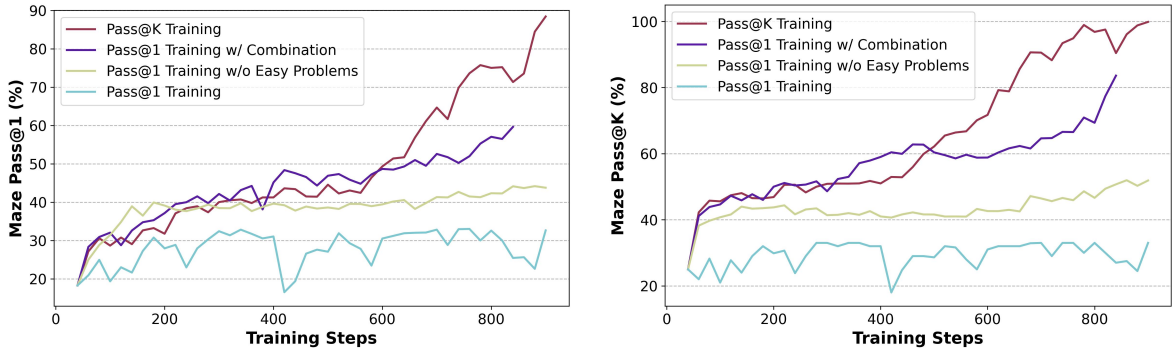
Figure 14: The curves of the advantage function on the setting of $N_{\text{rollout}} = 32$.

ments and present the results in Table 12. We can observe that Pass@1 and Pass@k Training can enhance the model's capacities on in-domain and out-of-domain tasks, suggesting the strong generalization ability of the RLVR training process. Further, comparing the performance between these two training approaches, the model trained through Pass@k outperforms the model trained on Pass@1. The reason behind it is that Pass@k Training encourages models to explore better solutions, which can be easily generalized to other tasks. In contrast, Pass@1 Training makes LLMs behave conservatively, thereby affecting LLMs' OOD performance.

Takeaway from Appendix I.2. Pass@k Training exhibits stronger generalization ability than Pass@1 Training, achieving greater improvements over the base model in both in-domain and out-of-domain testing.

I.3 Whether the Pass@k Performance can be Improved More Rapidly?

In previous discussion, we have found that the position of maximum value of ηwill influence the training objective (focus on Pass@1 or Pass@k). Based on these observations and conclusions, we hypothesize that an earlier peak in the η function leads to better optimization performance in Pass@k



(a) Pass@1 Performance of Maze Tasks.

(b) Pass@k Performance of Maze Tasks.

Figure 15: Training progress of various of Pass@1 Training and Pass@k Training.

Table 10: Pass@k performance of Enigmata of different RLVR methods. "P@k T." denotes the Pass@k Training with analytical derivation.

	Crypto	Arithmetic	Logic	Grid	Graph	Search	Sequential	Overall
Qwen2.5-7B-Ins	0.7	3.3	48.4	9.2	11.6	1.0	5.4	10.1
+ DAPO	5.7	28.0	68.4	22.8	20.4	6.8	12.7	21.3
+ DAPO & P@k T.	39.7	63.0	74.0	27.8	21.5	12.3	18.3	**29.8 (+8.5)**
+ GRPO	0.3	13.0	65.6	19.4	14.4	2.6	12.4	17.3
+ GRPO & P@k T.	0.7	17.7	64.2	20.8	18.2	4.0	12.7	**18.6 (+1.3)**
+ ForkingToken	94.0	35.0	67.1	25.9	21.6	13.4	11.7	29.1
+ ForkingToken & P@k T.	91.3	48.3	67.6	27.4	27.8	22.3	18.6	**34.0 (+4.9)**

Training. To test this hypothesis, we design a transformation function as follows:

$$f(N_{\text{pos}}) = \frac{4}{10\log(N_{\text{pos}} + 0.5)}, \quad \hat{A}' = f(N_{\text{pos}}) \times \hat{A}. \tag{25}$$

The advantage value curve after applying the transformation function is shown in Fig. 14e. We observe that, in the transformed curve, the peak of the η function is shifted forward to the position where the correctness is $\frac{1}{32}$. According to our hypothesis, such a modification of the advantage function is expected to result in better optimization performance for Pass@k Training. We integrate this transformed function into the RLVR training process (named as *Exceeding Pass@k Training*), and the corresponding training results are presented in Figure 16.

From the experimental results, we observe that Exceeding Pass@k Training can effectively improve the model's Pass@k performance during the early training stage. However, since this method places excessive emphasis on difficult problems, the improvement in Pass@1 performance on downstream tasks progresses more slowly. Based on these observations and analyses, we hypothesize that the computation of advantage values could be adaptively adjusted according to the model's current state. We leave this as a direction for future work.

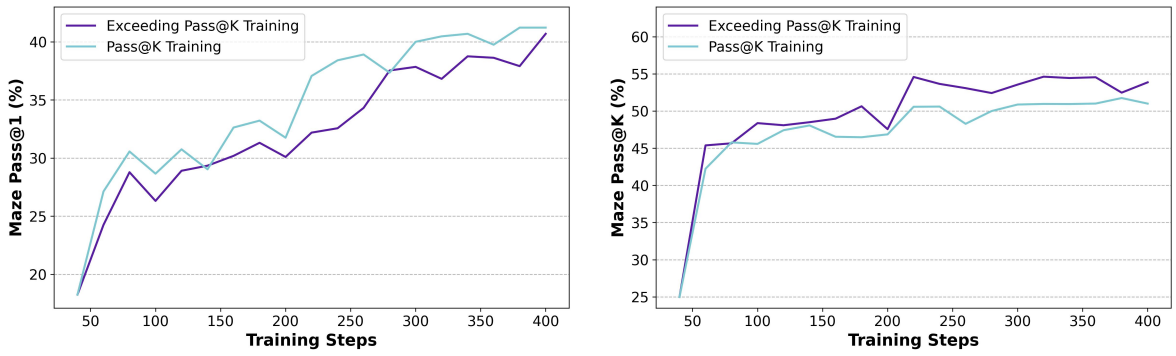
Takeaway from Appendix I.3. By specifically designing the advantage function, the model's Pass@k performance can be rapidly improved, enabling a fast enhancement of model capability during the early stage of training and further demonstrating the effectiveness and potential of implicit reward design.

I.4 Will Rollout Number Affect the Training Effectiveness?

We present the model's Pass@1 scores, Pass@k scores, and entropy under different numbers of rollouts in Table 1. From the above experimental results, it can be observed that Pass@k Training achieves improvements across different numbers of rollouts.At the same time, we observe that reducing the number of rollouts does not lead to significant fluctuations in model performance, indicating that our method is highly robust. For the token-level entropy of the model, we observe that models trained with Pass@k Training exhibit higher entropy, which can be reduced through further Pass@1 training, transferring the enhanced

Table 11: Pass@1/Pass@k Performance of Qwen2.5-7B-Instruct trained on different approaches.

Pass@1/Pass@k	**In-Domain Tasks**		**Out-of-Domain Tasks**	
	ARC-AGI 1	Enigmate	KORBench	AIME 2025
Qwen2.5-7B-Instruct	2.4/4.8	4.8/10.1	36.5/45.9	4.2/15.8
+ Pass@1 Training	3.3/3.8	12.9/21.3	37.7/45.6	5.4/19.1
+ Pass@k Training	**4.0/5.3**	**17.9/29.8**	**47.7/63.5**	**7.1/22.4**



(a) Pass@1 Performance of Maze Tasks.

(b) Pass@k Performance of Maze Tasks.

Figure 16: Training progress of Pass@k Training and Exceeding Pass@k Training.

Table 12: Pass@1/Pass@k score and Entropy of Qwen2.5-7B-Ins trained on different approaches on mathematical tasks.

Methods	**Pass@1/Pass@k**	**Entropy**
	$N_{\text{rollout}} = 16$	
P@1 T.	20.9/47.1	0.55
ForkingToken	21.1/46.9	0.52
EntropyAdv	19.8/49.8	0.61
P@k T.	20.7/51.3	0.67
P@k T. + P@1 T.	24.1/54.3	0.58
	$N_{\text{rollout}} = 8$	
P@1 T.	20.3/46.0	0.53
ForkingToken	21.1/45.7	0.54
EntropyAdv	19.3/49.3	0.59
P@k T.	20.5/50.8	0.60
P@k T. + P@1 T.	23.5/54.0	0.52

exploration capability to exploitation capability.

Takeaway from Appendix I.4. Pass@k Training consistently improves performance across rollout numbers, showing strong robustness while enhancing exploration capability that can be transferred to exploitation through subsequent Pass@1 training.

I.5 Discussion about Other Entropy-guided Approaches.

We compare the effectiveness between Pass@k Training and the naive implementation of Entropy-guided Approach (*i.e.,* Entropy Regularization). Moreover, there are several other methods, such as integrating the entropy into the advantage function (Cheng et al., 2025) or focusing on tokens with high covariance (Cui et al., 2025). Similarly, these methods might introduce a new trade-off: overly strict constraints may lead to underfitting and insufficient model training, while overly loose constraints can result in instability during training, potentially affecting the training effectiveness and model performance (He et al., 2025; Hong et al., 2025; Casper et al., 2023), since entropy conflicts with the Pass@1 metric. Therefore, the hyper-parameters should be carefully selected during the above methods to bring the performance improvement of LLMs. Actually, these methods are orthogonal to Pass@k Training, meaning that we can also combine these methods with Pass@k Training to achieve better training results. To verify this, we conduct the experiments in Sec. 5 to assess the effectiveness of Pass@k Training based on the guidance of policy entropy, demonstrating significant improvements.