SPS: Steering Probability Squeezing for Better Exploration in Reinforcement Learning for Large Language Models

Yifu Huo[1], Chenglong Wang[1], Ziming Zhu[1], Shunjie Xing[1], Peinan Feng[1], Tongran Liu[2], Qiaozhi He[1] Tianhua Zhou[3], Xiaojia Chang[3], Jingbo Zhu[1], Zhengtao Yu[4], and Tong Xiao[1]*

[1]Northeastern University, Shenyang, China
[2]CAS Key Laboratory of Behavioral Science, Beijing, China
[3]Independent Researcher, Beijing, China
[4]Kunming University of Science and Technology, Kunming, China
ifnoct@gmail.com xiaotong@mail.neu.edu.cn

Abstract

Reinforcement learning (RL) has emerged as a promising paradigm for training reasoning-oriented models by leveraging rule-based reward signals. However, RL training typically tends to improve single-sample success rates (*i.e.*, Pass@1) while offering limited exploration of diverse reasoning trajectories, which is crucial for multi-sample performance (*i.e.*, Pass@k). Our preliminary analysis reveals that this limitation stems from a fundamental *squeezing effect*, whereby probability mass is excessively concentrated on a narrow subset of high-reward trajectories, restricting genuine exploration and constraining attainable performance under RL training. To address this issue, in this work, we propose **S**teering **P**robability **S**queezing (SPS), a training paradigm that interleaves conventional RL with inverse reinforcement learning (IRL). SPS treats on-policy rollouts as demonstrations and employs IRL to explicitly reshape the induced trajectory distribution, thereby enhancing exploration without introducing external supervision. Experiments on five commonly used reasoning benchmarks demonstrate that SPS can enable better exploration and improve Pass@k. Beyond algorithmic contributions, we provide an analysis of RL learning dynamics and identify an empirical upper bound on Pass@k, shedding light on intrinsic exploration limits in RL-based reasoning models. Our findings suggest that alternating between RL and IRL offers an effective pathway toward extending the exploration capacity of reasoning-oriented large language models.

1 Introduction

In recent years, large language models (LLMs) have demonstrated impressive performance across a broad spectrum of foundational natural language processing (NLP) tasks, including text summarization, dialogue systems, and machine translation (Stiennon et al., 2020; Wang et al., 2024a; Luo et al., 2025). Building on the advances, the research community has increasingly shifted its focus toward more challenging research frontiers, especially in reasoning and code generation (Lightman et al., 2024; Li et al., 2025), and has even begun exploring the use of LLMs in the discovery of novel scientific theorems (Georgiev et al., 2025). As a result, exploration has emerged as a key capability of LLMs for future progress in these domains.

Motivated by the growing importance of exploration in reasoning-centric applications, contemporary LLM alignment methods have begun to explicitly incorporate exploration into the training pipeline. A simple and widely adopted strategy is to draw multiple samples per prompt to obtain a diverse set of candidate responses, where the model's exploration capability is essential for ensuring output diversity (Liu et al., 2024; Wang et al., 2024b). However, such multi-sample strategies merely increase surface-level diversity by repeatedly sampling from an unchanged policy, without fundamentally enhancing the entropy of the underlying distribution, resulting in highly inefficient exploration (Cui et al., 2025).

This limitation has been further substantiated by recent empirical studies. For example, Yue et al. (2025) demonstrate that although RL training substantially improves Pass@1 under large-scale sampling, the corresponding gains in Pass@k grow much more slowly, reflecting insufficient exploration of alternative reasoning trajectories. In essence, RL primarily improves sampling efficiency to boost single-sample success rates, rather than uncovering diverse trajectories that would meaningfully enhance multi-sample performance. To mitigate this sharpening effect and promote exploration, recent work has extended vanilla RL methods primarily along a common direction: explicitly counteracting entropy collapse to encourage broader exploration during RL training (Liu et al., 2025; Cui et al., 2025).

*Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 17472–17489
July 2-7, 2026

In this work, we advance this line of research by investigating a fundamental *squeezing effect* in RL training (Ren and Sutherland, 2024). This effect characterizes a systematic bias in probability mass redistribution. Specifically, negative gradients applied to low-probability responses fail to reallocate probability mass toward positively reinforced alternatives; instead, the removed mass is disproportionately absorbed by the greedy (*i.e.*, already dominant) response. As a consequence, the output distribution becomes increasingly concentrated, exacerbating distributional sharpening rather than promoting exploration. Our preliminary analysis reveals that this squeezing effect constitutes an intrinsic limitation of exploration in RL-based training. Moreover, we provide a theoretical justification supporting this insight, formalizing how probability mass redistribution under standard RL objectives leads to progressive concentration (Please refer to Appendix A).

Motivated by this analysis, we aim to explicitly enhance exploration by mitigating the squeezing effect. To this end, we propose **S**teering **P**robability **S**queezing (SPS), an RL training approach that extends conventional RL by interleaving inverse reinforcement learning (IRL) stages. Our basic idea is that, following standard RL training, we employ an IRL to explicitly reshape the induced trajectory distribution, reallocating probability mass away from overly dominant responses toward under-explored but potentially valuable alternatives. Specifically, compared to vanilla RL, SPS periodically incorporates forward IRL updates (Sun and van der Schaar, 2024), using only on-policy rollouts as demonstrations to avoid introducing external supervision or prior knowledge. Additionally, to further enhance exploration, we design an iterative SPS training strategy that repeatedly alternates between RL and IRL updates, enabling progressive redistribution of probability mass and preventing premature concentration of the policy.

Our core contributions are threefold:

- We conduct a preliminary analysis of the training dynamics in RL and identify an empirical upper bound on Pass@k. Our analysis results reveal the presence of a *squeezing effect* in RL, which constrains exploration.
- Building on this analysis, we propose the SPS approach, which employs IRL to explicitly reshape the induced trajectory distribution, thereby facilitating enhanced exploration. Additionally, we introduce an iterative SPS training strategy to further enhance exploration.
- We evaluate SPS on five Olympiad-level mathematical benchmarks. The experimental results demonstrate consistent and substantial improvements in Pass@k, indicating that SPS effectively broadens exploration and facilitates the discovery of diverse reasoning trajectories. Notably, on the Qwen2.5-Math-1.5B model, SPS achieves a Pass@128 score of 63.33 on the BRUMO benchmark, representing an improvement of +10.00 points compared to the vanilla GRPO (Shao et al., 2024).

2 Preliminaries

2.1 Task Formulation

Enhancing LLM Reasoning via Constrained Data. Given a finite set of reasoning questions x with corresponding ground truth label l, the objective of enhancing LLM reasoning is to learn a policy that produces correct reasoning trajectories through on-policy rollouts. During training, the policy iteratively samples multiple trajectories and receives corresponding outcome-level feedback extracted from the validator. The validator can be written as

$$R(y, l) \quad = \quad \mathbb{I}[v(y) = l] \tag{1}$$

where $v(\cdot)$ denotes an extraction function that extracts the answer from response y. In mathematical reasoning, the validator is commonly formulated as an indicator function, assigning a value of 1 when the extracted answer exactly matches the ground truth l, and 0 otherwise.

Exploration on Reasoning Tasks. In LLMs training, exploration refers to the ability of a learning process to expand the set of correct reasoning trajectories rather than simply reweighting partial existing patterns. Formally, given a base policy $\pi_{\text{base}}(\cdot)$ and a training policy $\pi_\theta(\cdot)$, exploration occurs if $\pi_\theta(\cdot)$ raises the probability to correct reasoning trajectories that are outside the high-likelihood region, thereby enlarging the boundary of the set of solvable problems.

Measurement of Exploration. Under our definition, effective exploration corresponds to expanding the set of problems that the model can successfully solve. To operationalize this notion, we adopt Pass@k as an estimation of the exploration.

Pass@k is commonly defined as the expected maximum reward obtained from k independently sampled responses for a given problem (Chen et al., 2025). Formally, it is computed as

$$\kappa_k = \mathop{\mathbb{E}}_{\substack{(x,l)\sim D \\ \{\hat{y}_i\}_{i=1}^{k}\sim\pi_\theta(\cdot|x)}} \Big[\max(R(\hat{y}_1,l), R(\hat{y}_2,l), \cdots, R(\hat{y}_k,l)) \Big] \tag{2}$$

where k is typically set to a relatively large value to reflect the model's exploration capability. Following prior studies (Ji et al., 2025), we set $k = 128$ throughout our experiments.

2.2 Group Relative Policy Optimization

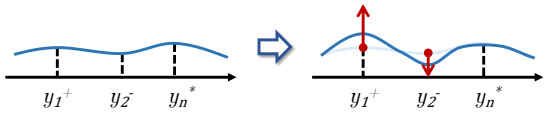
GRPO has emerged as one of the most widely adopted RL algorithms for training LLMs. Compared to standard PPO (Schulman et al., 2017), GRPO estimates advantages using a group of G rollouts rather than relying on a separate value network. Despite this multi-sample formulation, the reward signal in the RLVR setting is binary (i.e., correct or incorrect), which allows the learning objective to be reformulated in a contrastive learning framework. Building on this observation, Wu et al. further decomposes the original objective into the following contrastive form:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \sqrt{\text{Var}(x)} \left(\mathop{\mathbb{E}}_{y^+\sim\pi_\theta^+(\cdot|x)} \frac{\pi_\theta(y^+|x)}{|y^+|} - \mathop{\mathbb{E}}_{y^-\sim\pi_\theta^-(\cdot|x)} \frac{\pi_\theta(y^-|x)}{|y^-|} \right) \tag{3}$$

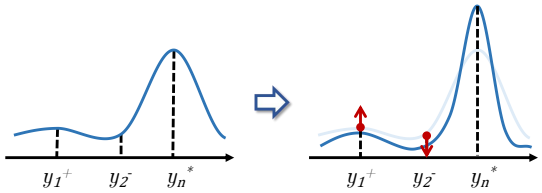
where $\text{Var}(\cdot)$ denotes the variance of the Bernoulli reward scores estimated from grouped samples, and y^+ and y^- denote positively and negatively rewarded samples, respectively. $\pi_\theta^+(\cdot)$ and $\pi_\theta^-(\cdot)$ denote the positive and negative policy, respectively.

3 Preliminary Analysis

Motivated by learning dynamics analyses (Ren and Sutherland, 2024), we hypothesize that the under-exploration issue in RL arises from an inherent squeezing effect induced by contrastive reward optimization. To validate this hypothesis, we conduct a two-stage analysis. First, we characterize how the squeezing effect emerges during RL training. Second, we explore how this effect restricts genuine exploration in reasoning tasks.



(a) Influence of gradients on a balanced distribution.



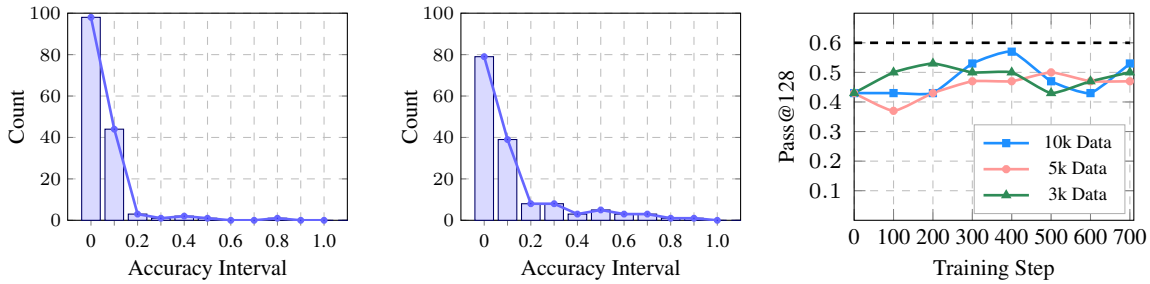
(b) Influence of gradients on a peaky distribution.

Figure 1: Illustration of *squeezing effect*. y_n^* denotes the sequence that dominates the output distribution (*i.e.*, the sequence consistently sampled by greedy decoding). Subfigure (a) shows the normal RL case, where probability mass shifts along the gradient direction. Subfigure (b) shows that when the distribution is already imbalanced, the updates further concentrate probability mass into the dominant peak, a phenomenon referred to as the *squeezing effect*.

3.1 Emergence of the Squeezing Effect in Reinforcement Learning

The *squeezing effect* describes a phenomenon in which applying negative gradient updates to low-probability tokens paradoxically causes the model's output distribution to concentrate further on the most likely token. As illustrated in Figure 1(a), when a policy model is trained with RL, its updates are jointly influenced by two opposing gradient components arising from the objective. Intuitively, the positive gradient increases the likelihood of positively rewarded samples, while the negative gradient suppresses the likelihood of negatively rewarded ones. However, this intuition breaks down under highly imbalanced output distributions, as shown in Figure 1(b). When a small number of tokens already dominate the distribution, the probability mass removed from low-probability tokens is not redistributed evenly; instead, it is effectively *squeezed* toward the dominant tokens, further amplifying their probabilities.

In fact, this counterintuitive behavior arises from the normalization property of the softmax function used in the model (Ren and Sutherland, 2024). Specifically, when a negative update is applied to a token with negligible probability, the token itself is barely affected. Instead, the update primarily increases the softmax normalization constant, which reduces the normalized probabilities of nearly all tokens. For tokens that already dominate the distri-



(a) Before GRPO (b) After GRPO (c) AIME-25

Figure 2: Partial results of the preliminary study. Subfigures (a) and (b) show the effect of GRPO on average question accuracy over the combined dataset. Subfigure (c) presents the dynamics of the Pass@128 metric during training, revealing an empirical boundary on exploration. More results can be found in Figure 5.

bution, however, this reduction is minimal in relative terms, causing their normalized probabilities to increase proportionally. As a result, probability mass progressively concentrates on the most likely token, leading to systematic sharpening of the output distribution and reduced diversity. A detailed theoretical proof of the squeezing effect is provided in Appendix A.

3.2 Impact of the Squeezing Effect on Exploration

In this subsection, we analyze the impact of the squeezing effect on RL performance from an exploration perspective. Inspired by recent studies highlighting the importance of entropy and distributional sharpness in RL (Cui et al., 2025; Yue et al., 2025), we argue that as the squeezing effect progressively reallocates probability mass toward already dominant tokens, the model's output distribution becomes increasingly concentrated. A more closely related phenomenon is reported by Tang et al., who observe that penalizing low-probability tokens suppresses unlikely outputs, thereby narrowing the distribution and reducing response diversity. This gradual loss of diversity directly constrains exploratory behavior during training, limiting the model's ability to discover alternative and potentially superior reasoning trajectories.

Based on this insight, we conduct a preliminary study focusing on the evolution of solvable questions during RL training. Specifically, we finetune Qwen2.5-Math-7B on 10k questions sampled from `Openr1-Math-46k-8192` using GRPO, and evaluate intermediate checkpoints on a combined benchmark consisting of the five Olympiad-level datasets. For each question, we compute the average pass rate across multiple sampled responses and discretize these values into accuracy buckets, enabling us to examine how performance is distributed throughout the course of GRPO training. Figures 2(a) and (b) present the histograms of the average Pass@1 accuracy distributions for the base model (denoted as *Before GRPO*) and the best GRPO checkpoint (denoted as *After GRPO*), respectively. As shown in the results, although GRPO introduces explicit exploration during training, the model does not consistently discover better trajectories for all questions. To further substantiate this observation, we also report Pass@128 results on AIME-25, where model checkpoints are evaluated every 100 training steps under different training data scales (3k, 5k, and 10k questions). Across all settings, we observe that increasing training steps does not lead to a monotonic improvement in Pass@128 performance, indicating that higher-quality trajectories are not continuously uncovered during training. Recent studies often attribute this phenomenon to entropy collapse in RL (Cui et al., 2025). However, rather than stopping at this surface-level explanation, we here probe a deeper underlying cause: *the probability squeezing effect, which naturally acts as a key mechanism that can trigger entropy collapse*.

4 Steering Probability Sequeezing

From our preliminary analysis, we have established two key findings: 1) the squeezing effect occurs in RL, and 2) this squeezing effect limits the exploration. These findings suggest that if we can steer this probability squeezing in a way that favors exploration, we could achieve improved RL performance. To this end, we propose an SPS approach, which explicitly steers the probability squeezing phenomenon via interleaving on-policy RL with inverse RL. The basic idea of SPS is to "redirect" the misallocated probability mass during squeezing: in-

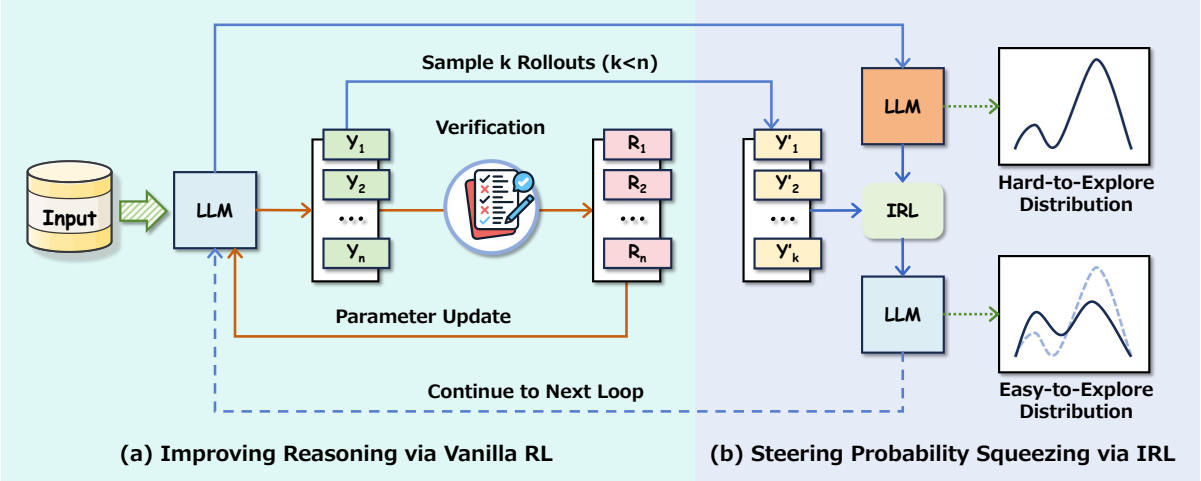


Figure 3: An overview of our SPS approach. Our training pipeline follows an iterative loop consisting of two complementary phases: (a) We first perform standard RL to explore the dataset and generate rollouts, from which a subset is sampled as demonstrations; (b) In the subsequent IRL phase, these demonstrations are leveraged to steer probability squeezing by reshaping the policy distribution. In practice, the two phases are interleaved to form a continual and unified training process.

stead of allowing it to converge to dominant greedy trajectories, we guide it toward under-explored regions that may contain better correct trajectories. The overview of SPS is shown in Figure 3. We present the details of SPS in the following sections.

4.1 Inverse Reinforcement Learning for Probability Redistribution

Standard RL typically leads to probability squeezing, where excessive mass concentrates on a narrow set of high-reward trajectories. In this work, we adopt IRL as a principled mechanism to steer probability redistribution by matching desired occupancy patterns, rather than relying on ad hoc entropy regularization or reward reweighting. In the IRL phase, we employ a forward-KL objective to reshape the policy's output distribution. The loss function is defined as:

$$\mathcal{L}_{\text{IRL}} = -\mathbb{E}_{\substack{x\sim D \\ y'\sim Y'_x}} \text{KL}(\pi_{\text{rollout}}(y'|x)\|\pi(y'|x)) \quad (4)$$

where x is sampled from the training dataset D, and y' is sampled from a rollout set Y'_x, which is obtained by uniformly sampling from the responses generated during the vanilla RL phase. Here, π_{rollout} denotes the empirical distribution over rollout completions, while π represents the current policy. This objective encourages the policy to align with rollout-supported solution trajectories and thereby promotes broader exploration.

Crucially, during the IRL phases, we treat the rollouts generated by the current policy as the sole source of "expert trajectories". Note that in this process, no external supervision, annotations, or domain knowledge is introduced. As a result, SPS preserves the mass-covering nature of RL, while encouraging the model to explore beyond the narrow high-reward modes reinforced by standard RL.

Why Inverse Reinforcement Learning? IRL is no stranger to learn target distributions from demonstrations (Sun and van der Schaar, 2024; Sun, 2024). From a theoretical perspective, IRL enjoys an advantage that is particularly well-suited to our setting: it enables learning directly from example trajectories without explicitly specifying or constraining the policy through divergence-based regularization (e.g., KL-based approaches). In our scenario, the goal is to explicitly control the probability squeezing phenomenon, specifically, to encourage probability mass to be redistributed toward under-explored yet potentially correct trajectories. Intuitively, this corresponds to steering the squeezing behavior in Figure 2(b) to operate more like Figure 2(a), where probability mass is concentrated around diverse high-quality trajectories rather than collapsing onto a few dominant ones. By incorporating IRL, we can directly leverage sampled rollouts as demonstrations to reshape the policy distribution, explicitly counteracting misallocated probability mass during squeezing. This allows the model to preserve exploration while still benefiting from reinforcement signals.

Low-Likelihood Trajectory Emphasis. Based on the analysis in Section 3.1, we observe that

the squeezing effect primarily arises when optimization is dominated by negative samples with extremely low model likelihood. This observation suggests that explicitly increasing the influence of such low-likelihood solutions may help alleviate the squeezing phenomenon. Motivated by this insight, we propose Low-Likelihood Trajectory Emphasis (L2TE), a strategy that preferentially samples rollouts from trajectories with relatively low model likelihood. By amplifying the learning signal from these under-explored solutions, L2TE encourages broader exploration and counteracts excessive probability concentration. To ensure stable IRL training, we further augment each sampled batch with positive trajectories whenever the number of available negative samples is insufficient.

4.2 Iterative Reinforcement Learning

Since the IRL phase explicitly reshapes the model's output distribution, it alleviates excessive distributional sharpening and thereby re-enables exploration within a fixed dataset. To further promote sustained exploration, we design a continually looped training strategy, as illustrated in Algorithm 1. Specifically, we first fine-tune the base model using vanilla RL and collect the resulting rollouts. From these rollouts, we sample a small subset that balances exploration diversity and computational efficiency. The selected rollouts are then used to perform IRL on the reinforced policy, which reshapes the output distribution by redistributing probability mass away from overly dominant trajectories. This updated policy is subsequently fed back into the next RL phase. By iterating this RL–IRL loop, the model can continue to explore alternative solution trajectories even under constrained data conditions, progressively expanding the boundary of solvable problems rather than prematurely converging to a narrow set of greedy behaviors.

5 Experiments

5.1 Experimental Setups

Dataset and Models. Our experiments were conducted on Openr1-Math-46k-8192 (Yan et al., 2025), which was a curated subset of OpenR1-Math-220k (Face, 2025). This subset removed excessively long or erroneous generations, ensuring that all questions were solvable. From this dataset, we constructed subsets of different scales (3k, 5k, and 10k) via uniform random sampling. For the base models, we conducted experiments using pretrained checkpoints from Qwen2.5-Math series, including 1.5B and 7B (Yang et al., 2024).

Training Details. We implemented our method on top of `SWIFT`, using `vLLM` as the inference backend (Kwon et al., 2023). During the RL stages, we adopted a completion-level batch size of 128 and employed a reduced learning rate of 5e-7 to stabilize long-horizon exploration. Rollout generation was performed with a sampling temperature of 1.0, and we sampled 8 responses per prompt. Math-Verify[1] was used as the reward function without any additional format or length-based rewards. After the RL phase, we collected the generated rollouts and sampled three responses out of the eight completions for the IRL stages. To mitigate overfitting during IRL, we used a batch size of 512 and a learning rate of 5e-10. We performed four training steps per iteration to support extended exploration. All experiments were conducted on a cluster of 4×8 NVIDIA H100 GPUs. More experimental details can be found in Appendix C.

Evaluation. We implemented our SPS method on top of the GRPO algorithm, making GRPO (Shao et al., 2024) our primary baseline. Additionally, we compared our method against several representative RL approaches, including DAPO (Yu et al., 2025) and GSPO (Zheng et al., 2025). Each baseline was implemented following the recommended configurations reported in the corresponding papers. We evaluated our models across three challenging olympiad-level mathematical benchmarks to examine the boundary of solvable questions: AIME (MAA), BRUMO (BRUMO), and HMMT (HMMT). For AIME, we considered both the 2024 and 2025 editions 2024; 2025, and for HMMT, we evaluated both HMMT-FEB and HMMT-NOV. The evaluation was performed using `EvalScope`[2] (Team, 2024), together with the benchmark data released by Balunović et al.. We reported Pass@128 and the average of Pass@1 (Avg@128) for all benchmarks, generating model outputs with a sampling temperature of 0.7.

5.2 Main Results

We report Pass@128 and Avg@128 for the best checkpoints within 700 training steps. The best checkpoint is selected according to Avg@128, as

[1] https://github.com/huggingface/Math-Verify
[2] https://github.com/modelscope/evalscope

Method	Params.	Pass@128					Avg@128				
		AIME		BRUMO	HMMT		AIME		BRUMO	HMMT	
		24	25	DEF.	FEB.	NOV.	24	25	DEF.	FEB.	NOV.
OpenR1-3k											
Qwen2.5-Math	1.5B	46.67	46.67	43.33	23.33	40.00	4.30	3.10	4.09	0.34	3.15
+GSPO	1.5B	50.00	36.67	46.67	23.33	36.67	6.41	2.86	11.51	0.42	3.98
+DAPO	1.5B	43.33	43.33	56.67	23.33	33.33	6.64	2.11	10.70	0.34	4.04
+GRPO	1.5B	**43.33**	43.33	53.33	**20.00**	23.33	**4.56**	**2.76**	**5.00**	0.23	**3.02**
+SPS	1.5B	**43.33**	**46.67**	**63.33**	**20.00**	**36.67**	4.48	2.61	4.71	**0.36**	2.99
Qwen2.5-Math	7B	43.33	43.33	43.33	13.33	30.00	3.80	5.50	1.48	0.16	1.07
+GSPO	7B	63.33	46.67	50.00	26.67	36.67	16.12	8.75	12.16	1.33	4.92
+DAPO	7B	63.33	46.67	56.67	26.67	26.67	10.39	5.26	9.74	0.47	3.65
+GRPO	7B	**70.00**	**50.00**	50.00	**33.33**	36.67	15.31	**9.48**	**14.84**	**1.48**	4.97
+SPS	7B	**70.00**	**50.00**	**56.67**	30.00	**43.33**	**16.38**	8.49	13.85	1.04	**5.60**
OpenR1-5k											
Qwen2.5-Math	1.5B	46.67	46.67	43.33	23.33	40.00	4.30	3.10	4.09	0.34	3.15
+GSPO	1.5B	56.67	46.67	56.67	26.67	36.67	6.12	3.41	11.48	0.42	4.04
+DAPO	1.5B	53.33	33.33	50.00	33.33	40.00	7.45	1.62	10.86	0.44	3.67
+GRPO	1.5B	43.33	33.33	50.00	**26.67**	26.67	4.35	2.69	**4.61**	0.29	**3.18**
+SPS	1.5B	**50.00**	**53.33**	**56.67**	**26.67**	**40.00**	**4.77**	**2.94**	4.30	**0.34**	3.10
Qwen2.5-Math	7B	43.33	43.33	43.33	13.33	30.00	3.80	5.50	1.48	0.16	1.07
+GSPO	7B	66.67	50.00	50.00	26.67	40.00	16.25	9.85	12.68	1.02	5.34
+DAPO	7B	66.67	46.67	56.67	33.33	26.67	14.37	6.20	12.79	1.20	4.51
+GRPO	7B	**73.33**	46.67	**63.33**	**33.33**	**36.67**	**17.37**	7.66	**11.67**	**1.22**	**5.34**
+SPS	7B	63.33	**53.33**	60.00	**33.33**	33.33	8.91	**8.41**	7.60	0.60	2.29
OpenR1-10k											
Qwen2.5-Math	1.5B	46.67	46.67	43.33	23.33	40.00	4.30	3.10	4.09	0.34	3.15
+GSPO	1.5B	50.00	50.00	50.00	23.33	30.00	5.94	2.76	11.15	0.52	3.80
+DAPO	1.5B	50.00	43.33	50.00	30.00	36.67	6.07	2.05	9.69	0.47	3.93
+GRPO	1.5B	40.00	36.66	**56.67**	20.00	**33.33**	4.27	**2.84**	**4.35**	**0.39**	2.79
+SPS	1.5B	**56.67**	**50.00**	**56.67**	**30.00**	**33.33**	**9.48**	2.53	3.46	0.29	**3.20**
Qwen2.5-Math	7B	43.33	43.33	43.33	13.33	30.00	3.80	5.50	1.48	0.16	1.07
+GSPO	7B	70.00	46.67	56.67	26.67	40.00	15.60	9.48	13.13	1.33	4.82
+DAPO	7B	60.00	43.33	53.33	26.67	30.00	14.82	4.85	12.93	1.09	4.35
+GRPO	7B	**66.67**	**56.67**	50.00	30.00	33.33	**14.69**	**8.75**	**13.41**	**1.02**	**5.21**
+SPS	7B	63.33	53.33	**66.67**	**36.67**	**36.67**	13.10	8.17	10.05	0.86	4.56

Table 1: Performance comparison of RL methods across a set of reasoning benchmarks. Results are highlighted in bold when SPS outperforms vanilla GRPO, indicating enhanced exploration.

this metric reflects the convergence quality of RL training, as shown in Table 1. Our results demonstrate that SPS consistently outperforms all RL baselines on Pass@128, while maintaining comparable Avg@128 performance. This indicates that SPS improves both single-sample and multi-sample performance in a synchronized manner. Notably, SPS substantially increases Pass@128, implying that it effectively expands the exploration boundary. Remarkably, Qwen2.5-Math-1.5B achieves a Pass@128 score of 63.33 using only 3k training samples, highlighting the effectiveness of SPS in data-constrained settings.

The results also reveal an interesting pattern: *the impact of GRPO varies with model scale*, and this trend is consistent across different data regimes. GRPO reduces Pass@128 for the 1.5B model, while improving it for the 7B model. We hypothesize that this phenomenon is closely related to the base model's initial output distribution. Smaller models (*e.g.*, 1.5B) tend to overfit the training corpus, leading to a sharper distribution. GRPO aggravates this squeezing effect, thereby suppressing exploration. In contrast, larger models benefit from GRPO, which appears to enhance exploration by leveraging their richer internal knowledge.

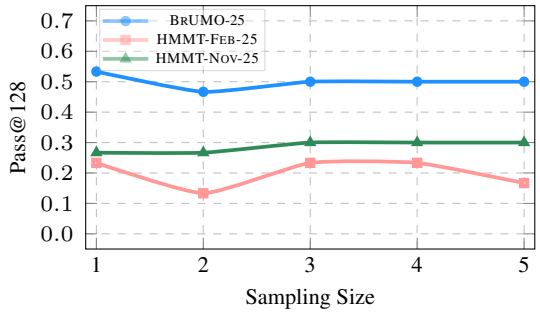


Figure 4: Impact of the sampling size on SPS performance. The experiments are conducted on the Qwen2.5-Math-1.5B model.

5.3 Ablation on Sampling Size

We further conducted an ablation study to investigate the effect of the sampling-size hyperparameter on training performance. Specifically, we applied SPS with varying sampling sizes on the 3k-sample dataset. Training was carried out for two epochs, and the results are summarized in Figure 4. The results demonstrate that model performance increased monotonically with larger sampling sizes. In practice, however, we balanced batch diversity against computational overhead and thus set the sampling size to three in all main experiments reported in this work.

6 Related Works

Reinforcement Learning for Large Reasoning Models. In the mainstream of current research, reasoning tasks are far from being low-hanging fruit. Unlike conventional NLP tasks, these *logic-intensive* problems require multi-step inference and strict logical consistency, making them substantially more difficult to solve. Interestingly, despite their inherent complexity, the correctness of final answers can often be easily validated through rule-based procedures, such as exact matching or program execution (Jiang et al., 2025; Xie et al., 2025; Huo et al., 2025; Wang et al., 2026). This property makes it feasible to train LLMs directly from outcome-level supervision, rather than relying on costly external annotations. Based on this observation, RL has emerged as an effective and explainable training paradigm for LLMs. Compared with RL from human feedback (RLHF), which relies on learned reward models to provide learning signals(Ouyang et al., 2022; Zhou et al., 2024; Wang et al., 2025a,b), RLVR replaces human preference annotations with deterministic validators, enabling scalable and low-cost reward generation. However, recent studies have indicated that RLVR suffers from degraded exploration, as the learning process tends to concentrate probability mass on a narrow set of high-reward solutions, leading to a sharpened output distribution and limited discovery of novel reasoning patterns (Yue et al., 2025).

Inverse Reinforcement Learning. IRL traditionally sought to infer an implicit reward function from expert demonstrations, framing learning as the recovery of objectives that rationalized observed behaviors (Sun and van der Schaar, 2024; Sun et al., 2024; Deng et al., 2024). In contrast to this classical setting, recent IRL-inspired approaches relaxed the reliance on external experts and instead operated on on-policy rollouts generated by the model itself (Wang et al., 2023). From a self-supervising perspective, the model's own trajectories serve as a proxy for demonstrations, allowing implicit reward functions to be extracted from its current behavior distribution (Zhang et al., 2021). Under this formulation, IRL no longer aims to exactly imitate an expert policy, but rather reshapes the reward or training signal to reweight model-generated trajectories, encouraging desirable solution patterns while preserving diversity. This perspective is particularly relevant in LLMs, where explicit rewards are often sparse or binary, and direct training tends to concentrate probability mass on a narrow set of high-reward outcomes. By leveraging on-policy rollouts as implicit supervision, IRL-style objectives provide a mechanism to smooth and redistribute the output distribution, complementing standard RL updates.

7 Conclusion

In this work, we have proposed SPS, an RL framework that interleaves on-policy RL with IRL to further enhance exploration. By learning from rollouts generated during the on-policy training phase, SPS can effectively mitigate the squeezing effect and significantly improve exploration compared with strong baselines across multiple olympiad-level reasoning benchmarks. These results underscore the critical role of IRL, which is often overlooked as current research primarily emphasizes purely RL-based training. Additionally, this work highlights the importance of analyzing RL from the perspective of learning dynamics, providing a clearer explanation of the behavior and limitations of existing training paradigms.

Limitations

While the proposed SPS approach provides a principled mechanism for steering probability mass to enhance exploration, several limitations warrant discussion. We discuss these limitations below:

- Although our experiments demonstrate practical effectiveness in reasoning tasks, the empirical validation is restricted to a relatively small set of models, with Qwen2.5-Math serving as the primary benchmark due to its consistently strong performance.
- Although probability steering proves effective in mitigating the *squeeze effect*, future work may explore more sophisticated mechanisms that more fully characterize and exploit the dynamics of reinforcement learning.
- Our current study does not analyze the inner states of policy models during training, leaving open questions regarding their interaction and relation to convergence behavior.

We acknowledge that we have not yet evaluated the method on larger-scale models. Due to computational constraints, our experiments focus on the 7B scale, which already allows us to study distributional concentration and exploration dynamics in a controlled setting.

Ethics Statement

This work does not need ethical considerations. The input of training is all from open-source data, and the output is also obtained based on open-source or commercial models.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Nos. U24A20334 and 62276056), the Yunnan Fundamental Research Projects (No.202401BC070021), the Yunnan Science and Technology Major Project (No. 202502AD080014), the Fundamental Research Funds for the Central Universities (Nos. N25BSS054 and N25BSS094), and the Program of Introducing Talents of Discipline to Universities, Plan 111 (No.B16009). We would like to thank the anonymous reviewers and SPC for their valuable comments, which helped improve this paper.

References

Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. 2025. Matharena: Evaluating llms on uncontaminated math competitions.

BRUMO. 2025. Brown university math olympiad 2025 (BrUMO).

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

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Hao-Si 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. *ArXiv preprint*, abs/2505.22617.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Jun-Mei Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiaoling Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 179 others. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645:633 – 638.

Zhirui Deng, Zhicheng Dou, Yutao Zhu, Ji-Rong Wen, Ruibin Xiong, Mang Wang, and Weipeng Chen. 2024. From novice to expert: Llm agent policy optimization via step-wise reinforcement learning. *ArXiv preprint*, abs/2411.03817.

Hugging Face. 2025. Open r1: A fully open reproduction of deepseek-r1.

Bogdan Georgiev, Javier G'omez-Serrano, Terence Tao, and Adam Zsolt Wagner. 2025. Mathematical exploration and discovery at scale. *ArXiv preprint*, abs/2511.02864.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Xiaodong Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *ArXiv*, abs/2009.03300.

HMMT. 2025. Harvard-mit mathematics tournaments (HMMT).

Yifu Huo, Chenglong Wang, Qiren Zhu, Shunjie Xing, Tong Xiao, Chunliang Zhang, Tongran Liu, and Jingbo Zhu. 2025. Heal: A hypothesis-based preference-aware analysis framework. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 8901–8919.

Xingguang Ji, Yahui Liu, Qi Wang, Jingyuan Zhang, Yang Yue, Rui Shi, Chenxi Sun, Fuzheng Zhang, Guorui Zhou, and Kun Gai. 2025. Leanabell-prover-v2: Verifier-integrated reasoning for formal theorem proving via reinforcement learning. *ArXiv preprint*, abs/2507.08649.

Xue Jiang, Yihong Dong, Mengyang Liu, Hongyi Deng, Tian Wang, Yongding Tao, Rongyu Cao, Binhua Li, Zhi Jin, Wenpin Jiao, Fei Huang, Yongbin Li, and Ge Li. 2025. Coderl+: Improving code generation via reinforcement with execution semantics alignment. *ArXiv preprint*, abs/2510.18471.

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

Dacheng Li, Shiyi Cao, Chengkun Cao, Xiuyu Li, Shangyin Tan, Kurt Keutzer, Jiarong Xing, Joseph Gonzalez, and Ion Stoica. 2025. S*: Test time scaling for code generation. *ArXiv preprint*, abs/2502.14382.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

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

Tianqi Liu, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter J. Liu, and Jialu Liu. 2024. Statistical rejection sampling improves preference optimization. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Yingfeng Luo, Tong Zheng, Yongyu Mu, Bei Li, Qinghong Zhang, Yongqi Gao, Ziqiang Xu, Peinan Feng, Xiaoqian Liu, Tong Xiao, and Jingbo Zhu. 2025. Beyond decoder-only: Large language models can be good encoders for machine translation. *ArXiv preprint*, abs/2503.06594.

MAA. 2025. American invitational mathematics examination (AIME). Mathematics Competition Series.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

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

Yi Ren and Danica J. Sutherland. 2024. Learning dynamics of llm finetuning. *ArXiv preprint*, abs/2407.10490.

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

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

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan J. Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2020. Learning to summarize from human feedback. *ArXiv preprint*, abs/2009.01325.

Hao Sun. 2024. Supervised fine-tuning as inverse reinforcement learning. *ArXiv preprint*, abs/2403.12017.

Hao Sun, Thomas Pouplin, Nicolás Astorga, Tennison Liu, and Mihaela van der Schaar. 2024. Improving llm generation with inverse and forward alignment: Reward modeling, prompting, fine-tuning, and inference-time optimization. In *The First Workshop on System-2 Reasoning at Scale, NeurIPS'24*.

Hao Sun and Mihaela van der Schaar. 2024. Inverselignment: Inverse reinforcement learning from demonstrations for llm alignment. *ArXiv preprint*, abs/2405.15624.

Xinyu Tang, Yuliang Zhan, Zhixun Li, Wayne Xin Zhao, Zhenduo Zhang, Zujie Wen, Zhiqiang Zhang, and Jun Zhou. 2025. Rethinking sample polarity in reinforcement learning with verifiable rewards.

ModelScope Team. 2024. EvalScope: Evaluation framework for large models.

Chenglong Wang, Yang Gan, Yifu Huo, Yongyu Mu, Qiaozhi He, Murun Yang, Bei Li, Tong Xiao, Chunliang Zhang, Tongran Liu, and 1 others. 2025a. Gram: A generative foundation reward model for reward generalization. *ArXiv preprint*, abs/2506.14175.

Chenglong Wang, Yang Gan, Yifu Huo, Yongyu Mu, Murun Yang, Qiaozhi He, Tong Xiao, Chunliang Zhang, Tongran Liu, and Jingbo Zhu. 2025b. Rovrm: A robust visual reward model optimized via auxiliary textual preference data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25336–25344.

Chenglong Wang, Yifu Huo, Yang Gan, Qiaozhi He, Qi Meng, Bei Li, Yan Wang, Junfu Liu, Tianhua Zhou, Jingbo Zhu, and 1 others. 2026. Msrl: Scaling generative multimodal reward modeling via multi-stage reinforcement learning. *ArXiv preprint*, abs/2603.25108.

Chenglong Wang, Hang Zhou, Kaiyan Chang, Bei Li, Yongyu Mu, Tong Xiao, Tongran Liu, and JingBo Zhu. 2024a. Hybrid alignment training for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11389–11403, Bangkok, Thailand. Association for Computational Linguistics.

Chenglong Wang, Hang Zhou, Yimin Hu, Yifu Huo, Bei Li, Tongran Liu, Tong Xiao, and Jingbo Zhu. 2024b. ESRL: efficient sampling-based reinforcement learning for sequence generation. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 19107–19115. AAAI Press.

Guojian Wang, Faguo Wu, Xiao Zhang, and Jianxiang Liu. 2023. Learning diverse policies with soft self-generated guidance. *International Journal of Intelligent Systems*, 2023(1):4705291.

Yihong Wu, Liheng Ma, Lei Ding, Muzhi Li, Xinyu Wang, Kejia Chen, Zhan Su, Zhanguang Zhang, Chenyang Huang, Yingxue Zhang, Mark Coates, and Jian-Yun Nie. 2025. It takes two: Your grpo is secretly dpo. *ArXiv preprint*, abs/2510.00977.

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. *ArXiv preprint*, abs/2502.14768.

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. 2025. Learning to reason under off-policy guidance. *ArXiv preprint*, abs/2504.14945.

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

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. *ArXiv preprint*, abs/2503.14476.

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

Linfeng Zhang, Chenglong Bao, and Kaisheng Ma. 2021. Self-distillation: Towards efficient and compact neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):4388–4403.

Yifan Zhang and Team Math-AI. 2024. American invitational mathematics examination (aime) 2024.

Yifan Zhang and Team Math-AI. 2025. American invitational mathematics examination (aime) 2025.

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

Hang Zhou, Chenglong Wang, Yimin Hu, Tong Xiao, Chunliang Zhang, and Jingbo Zhu. 2024. Prior constraints-based reward model training for aligning large language models. In *Proceedings of the 23rd Chinese National Conference on Computational Linguistics (Volume 1: Main Conference)*, pages 1395–1407.

Supplementary Materials for SPS

A Proofs for Theoretical Results

A.1 Derivation of the Squeezing Effect

Property 1. *The squeezing effect arises when negative gradient updates are applied to low-probability tokens, leading to a systematic sharpening of the model's output distribution.*

Proof: This behavior is inherent to the normalization structure of the softmax function. Let the model output distribution over the vocabulary be given by

$$p(i) = \frac{e^{z_i}}{Z}, \qquad Z = \sum_j e^{z_j} \tag{5}$$

where z_i denotes the logit associated with token i. Consider a token m that receives a negative logit update during training,

$$z_m \leftarrow z_m + \eta, \qquad \eta < 0 \tag{6}$$

which yields the updated distribution

$$p'(i) = \frac{e^{z_i}}{Z'}, \qquad Z' = e^{z_m+\eta} + \sum_{j\neq m} e^{z_j} \tag{7}$$

For any token $j \neq m$, we may express the updated probability in terms of the original distribution as

$$p'(j) = \frac{p(j)}{1 + p(m)\,(e^{\eta} - 1)} \tag{8}$$

The squeezing effect typically arises when the distribution satisfies $p(m) \ll 1$. Performing a first-order expansion then gives

$$p'(j) \approx p(j)\,[1 - p(m)\,(e^{\eta} - 1)] \tag{9}$$

Since $\eta < 0$ implies $e^{\eta} - 1 < 0$, it follows that

$$p'(j) < p(j), \qquad \forall j \neq m \tag{10}$$

That is, the normalized probabilities of almost all tokens decrease simultaneously. However, the best-probability token, being the dominant term in the distribution, which experiences the smallest relative decrease, implying

$$\max_i p'(i) > \max_i p(i) \tag{11}$$

Thus, the probability mass is progressively concentrated toward the most likely token, and the output distribution becomes increasingly peaked. This phenomenon is referred to as the squeezing effect (Ren and Sutherland, 2024).

A.2 Squeezing Effect at the Sequence Level

The previous analysis establishes that penalizing low-probability tokens induces probability mass to concentrate toward the modal token due to the normalization structure of the softmax function. We now generalize this reasoning to sequence-level probability distributions, which are central to policy optimization in language model training.

Property 2. *The squeezing effect arises when negative gradient updates are applied to low-probability sequences, leading to a systematic sharpening of the model's output distribution.*

Proof: Let a sequence be denoted by

$$y = \{y_1, ..., y_T\} \tag{12}$$

and let the model define the joint probaiblity

$$p(y) = \prod_{t=1}^{T} p(y_t \mid y_{<t}) \tag{13}$$

where each conditional distribution is parameterized by a softmax over logits z_t. Suppose that a particular sequence y^- receives a negative gradient update under the training objective, effectively reducing its log-probability. This corresponds to a logit-space update of the form

$$\log p(y^-) \leftarrow \log p(y^-) + \eta, \qquad \eta < 0 \tag{14}$$

At the sequence level, the normalized model distribution over all candidate sequences $\mathcal{Y}$ may be represented as

$$P(y) = \frac{\exp(\log p(y))}{\sum_{y' \in \mathcal{Y}} \exp(\log p(y'))} \tag{15}$$

After the update to y^-, the new distribution becomes

$$P'(y) = \frac{\exp(\log p(y))}{\exp(\log p(y^-) + \eta) + \sum_{y' \neq y^-} \exp(\log p(y'))} \tag{16}$$

For any $y \neq y^-$, we obtain

$$P'(y) = \frac{P(y)}{1 + P(y^-)\,(e^{\eta} - 1)} \tag{17}$$

If the penalized sequence is already extremely unlikely, *i.e.*

$$P(y^-) \ll 1 \tag{18}$$

then a first-order expansion yields

$$P'(y) \approx P(y)\left[1 - P(y^-)\,(e^{\eta} - 1)\right] \tag{19}$$

Since $\eta < 0$ implies $e^{\eta} - 1 < 0$, it follows that

$$P'(y) < P(y), \qquad \forall y \neq y^-. \tag{20}$$

Thus, the normalized probability of nearly every sequence decreases simultaneously.
Let

$$y^\star = \arg\max_y P(y) \tag{21}$$

$y^\star$ denotes the most probable sequence. Because this sequence dominates the distribution, its relative decrease under normalization is smallest. Consequently,

$$\max_y P'(y) > \max_y P(y), \tag{22}$$

implying that probability mass becomes increasingly concentrated on $y^\star$.

B RLVR Algorithms

In this section, we enumerate the RLVR algorithms referred in this paper.

B.1 Group Relative Policy Optimization (GRPO)

In RLVR, GRPO has become one of the most widely used RL algorithms for LLM training. GRPO maximizes expected rewards by increasing the likelihood of higher-reward samples within a group, while normalizing each sample's advantage by the group's average reward and variance. It removes the critic network and instead computes a relative advantage inside each sampled group, then applies a PPO-style clipped objective to stabilize updates. The loss function of GRPO can be written as

$$\mathcal{J}_{\text{GRPO}}(\theta) = \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|y_i|}\sum_{t=1}^{|y_i|}\min\Big(w_i(\theta)\,\widehat{A}_i,\ \text{clip}(w_i(\theta),\,1-\varepsilon,\,1+\varepsilon)\,\widehat{A}_i\Big) \tag{23}$$

where $w_i(\theta)$ denotes an importance ratio, which can be computed as

$$w_i(\theta) = \frac{\pi_\theta(y_i \mid x)}{\pi_{\theta_{\text{old}}}(y_i \mid x)} \tag{24}$$

Specially, GRPO computes the advantages $\widehat{A}_i$ by normalizing rewards within a group of responses. In RLVR, we use outcome-level feedback given in Equation 1 as reward, therefore the advantages are computed as:

$$\widehat{A}_i \quad = \quad \frac{R_i - \text{mean}(\{R_1,\ldots,R_G\})}{\text{std}(\{R_1,\ldots,R_G\})} \tag{25}$$

B.2 Dynamic Sampling Policy Optimization (DAPO)

To stabilize RL training, Yu et al. (2025) propose DAPO. In DAPO, the clipping range is asymmetric: the lower bound remains restrictive to control instability, while the upper bound is relaxed to encourage exploration of low-probability tokens. Unlike GRPO, gradients are computed at the token level and averaged across all tokens in all sampled responses. Prompts for which all sampled responses are correct or all are incorrect are filtered out so that every retained prompt contributes a non-zero learning signal. The loss function of DAPO can be written as

$$\begin{aligned}\mathcal{J}_{\text{DAPO}}(\theta) = \frac{1}{\sum_{i=1}^{G}|y_i|}\sum_{i=1}^{G}\sum_{t=1}^{|y_i|}\min\Big(r_{i,t}(\theta)\,\widehat{A}_i,\ \text{clip}\big(r_{i,t}(\theta),\,1-\varepsilon_{\text{low}},\,1+\varepsilon_{\text{high}}\big)\widehat{A}_i\Big),\\ \text{s.t.}\quad 0 < \Big|\{y_i \mid R(y_i,l)=1\}\Big| < G\end{aligned} \tag{26}$$

where $r_{i,t}(\theta)$ denotes a token-level importance ratio, as follows:

$$r_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t} \mid x, y_{i<t})}{\pi_{\theta_{\text{old}}}(y_{i,t} \mid x, y_{i<t})} \tag{27}$$

B.3 Group Sequence Policy Optimization (GSPO)

GSPO optimizes a sequence-level clipped objective, where each response's normalized reward (advantage) is weighted by its sequence-likelihood ratio between the current and old policy. In essence, it performs PPO-style clipping at the whole-sequence level, aligning off-policy correction and optimization with the sequence-level reward. The loss function of GSPO can be written as

$$\mathcal{J}_{\text{GSPO}}(\theta) = \frac{1}{G}\sum_{i=1}^{G}\min\Big(s_i(\theta)\,\widehat{A}_i,\ \text{clip}(w_i(\theta),\,1-\varepsilon,\,1+\varepsilon)\,\widehat{A}_i\Big) \tag{28}$$

while its importance ratio $s_i(\theta)$ is differently computed as

$$s_i(\theta) = \left(\frac{\pi_\theta(y_i|x)}{\pi_{\theta_{\text{old}}}(y_i|x)} \right)^{\frac{1}{|y_i|}} = \exp\left(\frac{1}{|y_i|} \sum_{t=1}^{|y_i|} \log \frac{\pi_\theta(y_{i,t}|x, y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t}|x, y_{i,<t})} \right) \tag{29}$$

Therefore, GSPO applies clipping to entire responses instead of individual tokens to exclude the overly "off-policy" samples from gradient estimation, which matches both the sequence-level rewarding and optimization (Zheng et al., 2025).

C Implementation Details

C.1 Iterative SPS

We propose the pseudo code of iterative SPS in Algorithm 1. The Iterative SPS algorithm iteratively enhances a base policy by alternating exploration and distribution reshaping: starting from the initial policy, it updates the policy on the dataset using vanilla RL to encourage exploration, collects grouped rollouts, samples a subset emphasizing low-likelihood or under-explored trajectories, and then applies IRL on this subset to reshape the policy distribution and mitigate probability squeezing, producing a more balanced and robust enhanced policy for exploration.

Algorithm 1: Iterative SPS

Input: Base policy $\pi_{\theta_0}(\cdot)$, dataset D, group size n, sampling size k
Output: Enhanced policy $\pi_\theta(\cdot)$
Initialize policy $\pi_\theta(\cdot) \leftarrow \pi_{\theta_0}(\cdot)$;
while *not converged* **do**
  `// Stage 1: Vanilla RL`
  Update $\pi_\theta(\cdot)$ on D using vanilla RL to encourage exploration;
  Collect grouped rollouts:

$$Y = \{\, y_x^1, \ldots, y_x^n \mid y_x^i \sim \pi_\theta(\cdot \mid x), x \in D \,\}$$

  `// PL2TE`
  Sample a subset $Y' \subset Y$, emphasizing low-likelihood or under-explored trajectories;
  `// Stage 2: IRL`
  Update $\pi_\theta(\cdot)$ via IRL on Y' by minimizing $\mathcal{L}_{\text{IRL}}$;
  `// Reshape the policy distribution to mitigate probability squeezing`
return π_θ;

In implementation, the rollout distribution π_{rollout} is instantiated through a degenerate discrete distribution over the sampled responses. Under this construction, the forward-KL objective can be reformulated in a manner that closely resembles a cross-entropy style penalty, as previously discussed in related literature (Sun, 2024). Interestingly, despite the apparent simplicity of this surrogate, empirical evidence suggests that it nevertheless facilitates an effective enlargement of the exploration region, even in the absence of explicit external guidance.

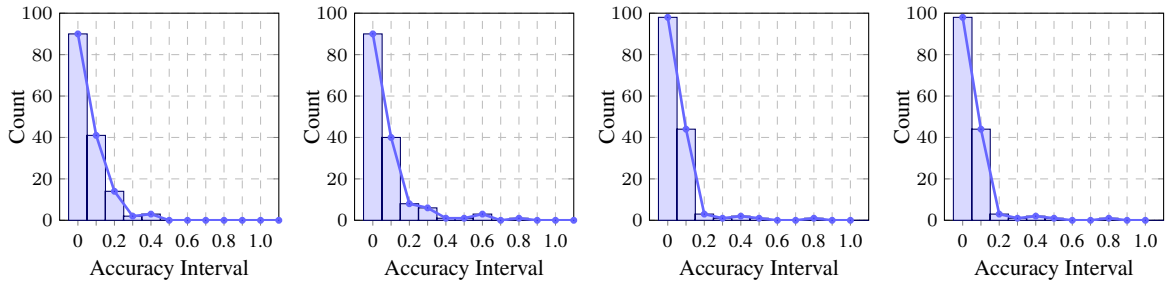
Furthermore, while any single low-probability response is unlikely to be sampled, the total number of such responses is exceedingly large. Consequently, the probability of obtaining at least one low-likelihood trajectory within a batch remains high. To better emphasize these low-probability trajectories, we sample from the lower quantile of trajectories in each batch.

C.2 Hyperparameter Setting

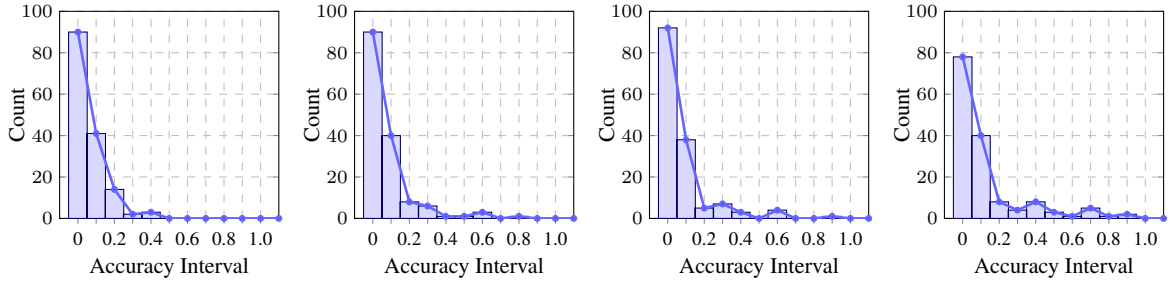
We conduct our main experiments using several RL algorithms, including GRPO (Shao et al., 2024), DAPO (Yu et al., 2025), and GSPO (Zheng et al., 2025). The algorithm-specific hyperparameters are

Method	Parameter Name	Value
GRPO	beta	0.01
	group_size	8
DAPO	epsilon_high	0.28
	max_resample_times	3
	soft_cache_length	2048
GSPO	beta	0.0
	epsilon	3e-4
	epsilon_high	4e-4
	steps_per_generation	4

Table 2: Hyperparameter setting of applied RL methods



(a)Accuracy distribution of Qwen2.5-Math-1.5B. From left to right: base model, GRPO, DAPO, and GSPO.



(b)Accuracy distribution of Qwen2.5-Math-7B. From left to right: base model, GRPO, DAPO, and GSPO.

Figure 5: Accuracy distribution variation of Qwen2.5-Math-1.5B

summarized in Table 2. For GRPO, we adopt a fixed group size across all baseline methods in order to balance the exploration capacity and computational cost. For DAPO and GSPO, we employ the recommended default configurations provided in `SWIFT`, which have been previously validated in practical deployments.

D Results on Other Backbone Models and Benchmarks

To further validate the generalizability of our method, we extend the main experiment to incorporate DeepSeek-R1 (DeepSeek-AI et al., 2025) as our additional backbone model. Furthermore, we extend our evaluation to other non-mathematical benchmarks, such as MMLU (Hendrycks et al., 2020) and GPQA-Diamond (Rein et al., 2023), the extended results are listed as Table 3

The extended results consistently show that SPS maintains its effectiveness across different model families and non-mathematical domains, supporting the general applicability of our method beyond competition-level mathematics.

Method	AIME-24	AIME-25	BRUMO	HMMT-FEB	HMMT-NOV	MATH-500	MMLU	GPQA-DIAMOND
DeepSeek-R1-Distill-Qwen-1.5B	63.33	36.67	60.00	33.33	36.67	72.39	44.33	14.14
+GRPO	70.00	46.67	**80.00**	40.00	36.67	83.42	44.86	19.70
+GSPO	73.33	**50.00**	66.67	30.00	40.00	76.51	44.16	21.21
+DAPO	66.67	40.00	60.00	33.33	26.67	83.46	**46.26**	**33.33**
+SPS	**80.00**	46.67	**80.00**	**46.67**	**43.33**	**84.25**	46.13	29.80

Table 3: Additional experimental results on DeepSeek-R1. The best results for each group are in **bold**. The second-best results for each group are with underline.

Stage	BASE	GRPO	SPS
Similarity	88.34	88.18	86.82

Table 4: Diversity comparison across different methods.

Stage	RL	IRL
Time(min)	68.10	2.25

Table 5: Time cost comparison across different stages.

E Further Analysis

E.1 Analysis of Diversity

As Pass@K (*e.g.*, Pass@128) is only an indirect proxy for exploration. Improvements in Pass@K may stem from multiple factors, including enhanced trajectory diversity. Nevertheless, prior work (Yue et al., 2025) suggests an intrinsic relationship between exploration dynamics and Pass@K metrics, as broader policy support generally increases the probability of sampling correct reasoning paths within a finite budget. In this sense, while Pass@K is not a direct diversity measure, it remains behaviorally correlated with exploration capacity.

At the same time, we fully agree that explicit diversity metrics provide more direct evidence. Following this suggestion, we compute trajectory-level similarity (lower indicates higher diversity), with results shown in Table 4.

The results show that SPS yields lower trajectory similarity, indicating increased reasoning diversity compared to both the base model and GRPO. This empirical evidence complements the Pass@K improvements and provides more direct support for our exploration claim.

E.2 A Cost-Benefit Analysis on SPS

As a multi-stage training method, SPS inevitably adds computational overhead and engineering complexity compared to vanilla RLVR methods. However, in practice, the IRL stage contributes only a negligible fraction of the total training time and does not introduce any substantial computational overhead. Empirically, the overall runtime is dominated by the RL rollout and policy optimization stage, whereas the IRL update introduces only marginal computational overhead. To quantify this, we measure the training time by training a 1.5B-parameter LLM on 3k prompts, isolating the respective stages. The measured training time is summarized in Table 5.

As shown, the IRL stage accounts for only about 3% of the total time per iteration, which is minor compared to the RL stage. Therefore, although the pipeline is conceptually multi-stage, the additional computational cost introduced by IRL is marginal and does not constitute a practical bottleneck.

E.3 Diagnostics for *Squeezing Effect*

We examine probability dynamics by collecting responses generated via greedy decoding and computing the average log-probability of the generated trajectories under the current policy. Intuitively, excessive probability squeezing manifests as over-concentration of mass on a narrow subset of trajectories, typically reflected in inflated log-probability magnitudes relative to the base model. We evaluate several algorithms on AIME 2024 and 2025, with results shown in Table 6.

Method	AIME-24	AIME-25
Base	-124.2745537	-120.7128742
GRPO	-122.4005783	-115.0455817
GSPO	-121.8923759	-115.0455817
SPS	-124.3440186	-115.0455817

Table 6: The average log-probability of the generated trajectories under different optimization methods.

Compared to GRPO and GSPO, SPS maintains log-probability levels much closer to the base model, indicating that it avoids aggressively concentrating probability mass on a small subset of trajectories. In contrast, GRPO and GSPO exhibit noticeably higher (less negative) log-probabilities, suggesting stronger probability squeezing.

These results provide direct empirical evidence that SPS mitigates probability squeezing while preserving performance gains. We will incorporate this diagnostic analysis into the revised manuscript to clarify the mechanism underlying SPS.

F Discussion

F.1 How SPS influence the reasoning?

SPS does not merely flatten the output distribution at the logit level. If its effect were equivalent to temperature scaling or entropy regularization, we would expect uniform entropy increases without meaningful changes in internal representations. However, SPS operates on trajectory-level objectives and reweights complete reasoning paths, which propagates gradients through intermediate transformer layers rather than only adjusting the final projection head. Empirically, the gains in high-K metrics exceed what would be predicted from Pass@1 improvements under an independent sampling assumption, suggesting reduced inter-sample redundancy rather than simple probability smoothing. Conceptually, post-hoc logit flattening cannot induce new reasoning modes, whereas SPS reshapes probability mass across semantically distinct trajectories.

F.2 Why the *Degenerate Discrete Distribution* works well?

The *degenerate discrete distribution* is simply the empirical distribution over RL rollouts, *i.e.*, a Monte Carlo estimator of the improved policy. Since the IRL stage only needs to match the relative structure within the sampled support, rather than reconstructing a continuous density as this empirical approximation is sufficient in practice. The target distribution is rollout-induced, so the empirical measure is a consistent surrogate. While a very small batch may increase variance, performance does not collapse in practice. This is partly due to the *rare-but-many* effect, although individual low-probability trajectories are hard to sample, their combinatorial cardinality is large, so typical batches still contain diverse underrepresented modes. Moreover, the IRL update is conservative (small learning rate), which prevents overfitting to sampling noise.