

RSPO: Risk-Seeking Policy Optimization for Pass@k and Max@k Metrics in Large Language Models

Kaichen Zhang^{1,2} Shenghao Gao^{1,2} Yuzhong Hong² Haipeng Sun² Junwei Bao²

Hongfei Jiang² Yang Song² Dingqian Hong² Hui Xiong¹

¹Hong Kong University of Science and Technology (Guangzhou)

²Zuoyebang Education Technology

Abstract

Current large language model post-training optimizes a *risk-neutral* objective that maximizes expected reward, yet evaluation relies heavily on *risk-seeking* metrics like Pass@k (at least one success in k trials) and Max@k (maximum reward across k responses). This mismatch in risk preferences can inevitably lead to suboptimal performance. To bridge this gap, we propose **Risk-Seeking Policy Optimization (RSPO)**, a novel method that directly targets Pass@k and Max@k during training. A key challenge in optimizing these metrics is the "hitchhiking" problem: low-reward responses are inadvertently reinforced if they co-occur with a high-reward response within a sample of k generations, resulting in inefficient optimization. RSPO addresses this problem by leveraging the closed-form probability that a given response is the maximum among k samplings. Despite the complexity of nested gradients over multiple responses, RSPO produces efficient, unbiased gradient estimators for both metrics. We validate our approach with both rigorous theoretical analysis and comprehensive experimental results.

CCS Concepts

- Computing methodologies → Machine learning algorithms; Natural language processing;
- Information systems → Data mining.

Keywords

Large Language Model, Post Training, Pass@k, Max@k

ACM Reference Format:

Kaichen Zhang^{1,2} Shenghao Gao^{1,2} Yuzhong Hong² Haipeng Sun² Junwei Bao², Hongfei Jiang² Yang Song² Dingqian Hong² Hui Xiong¹, ¹Hong Kong University of Science and Technology (Guangzhou), ²Zuoyebang Education Technology, . 2018. RSPO: Risk-Seeking Policy Optimization for Pass@k and Max@k Metrics in Large Language Models. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 13 pages. <https://doi.org/XXXXXXX.XXXXXXX>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Conference acronym 'XX, Woodstock, NY

© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-XXXX-X/2018/06

<https://doi.org/XXXXXXX.XXXXXXX>

Policy π_A :

100%: 6

Policy π_B :

50%: 0

50%: 10

Risk-neutral Training prefers π_A :

$E_{y \sim \pi_A} R(y) = 6 \text{ } \$ > E_{y \sim \pi_B} R(y) = 5 \text{ } \$$

Risk-seeking Evaluation prefers π_B :

$E_{y_1, y_2 \sim \pi_A} \max\{R(y_1), R(y_2)\} = 6 \text{ } \$$

$< E_{y_1, y_2 \sim \pi_B} \max\{R(y_1), R(y_2)\} = 7.5 \text{ } \$$

Figure 1: The risk preference mismatch between training and evaluation leads to different policy preferences.

1 Introduction

Large language models (LLMs) [15, 34] have demonstrated remarkable capabilities in natural language understanding and generation, driven by architectures such as the Transformer [29] and scaled-up pre-training [35] on massive text corpora. Post-training [28] techniques—such as supervised fine-tuning (SFT) [17] and most notably reinforcement learning (RL) [26] have emerged as a powerful paradigm to tailor models toward specific tasks and human preferences. By defining a reward function that captures desired behaviors (e.g., helpfulness, correctness), reinforcement learning can steer LLM outputs toward higher-quality responses and align them more closely with end-user needs [2].

Despite these successes, a fundamental mismatch persists between the objectives used during RL training and the metrics by which LLMs are evaluated. Standard reinforcement learning maximizes the expected reward across all generations—an inherently *risk-neutral* objective—whereas practical evaluation often hinges on *risk-seeking* metrics such as Pass@k (the probability that at least one of k samples is correct) or Max@k (the highest reward obtained among k sampled responses). For example, [25] employs an inference-time strategy that selects the optimal output from a set of generated candidates, while [33] uses Pass@k as a metric to assess the inherent reasoning capabilities of LLMs.

This divergence in risk preference can lead to suboptimal behavior: policies tuned to maximize average reward may under-explore high-variance, high-reward responses, resulting in degraded Max@k performance. Figure 1 demonstrates the risk preference mismatch, in which training prefers π_A but evaluation prefers π_B .

A critical obstacle to bridging this gap is the so-called "hitchhiking" phenomenon. When optimizing Pass@k or Max@k, low-reward responses that co-occur with a single high-reward sample within a batch of k can receive undue positive reinforcement, illustrated in Figure 2. Such hitchhiking skews the gradient signal away from genuinely promising responses, leading to inefficient learning and slower convergence toward optimal policies.

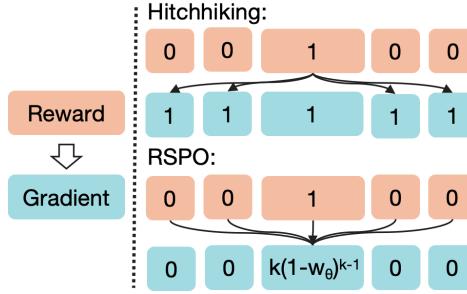


Figure 2: An illustration comparing "hitchhiking" baseline and RSPO for computing gradient weights from rewards.

To address these challenges, we propose Risk-Seeking Policy Optimization (RSPO), a novel RL algorithm that directly targets Pass@k and Max@k metrics.

RSPO derives a closed-form expression for the probability that any given response has the maximum reward in a set of k samples. This expression decouples the joint distribution of multiple responses in the Pass@k and Max@k objectives into a distribution over a single response, circumventing the "hitchhiking" issue.

Utilizing this expression, we derive a simplified objective for the Pass@k objective, wherein rewards are binary-valued:

$$\mathbb{E}_{x \sim \mathcal{D}} \left[\sum_y \pi_\theta(y|x) \frac{1 - (1 - w_\theta)^k}{w_\theta} R(x, y) \right]$$

In this objective, w_θ represents the probability that policy π_θ generates a reward of one. Although this formulation initially appears to involve nested gradients across multiple responses (due to the gradient term $\nabla_\theta(\pi_\theta(y|x) \frac{1 - (1 - w_\theta)^k}{w_\theta})$), we demonstrate that the gradient of the Pass@k objective can be elegantly simplified to:

$$\mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta(y|x)} \left[k(1 - w_\theta)^{k-1} R(x, y) \nabla_\theta \log \pi_\theta(y|x) \right]$$

The resulting gradient weight, $k(1 - w_\theta)^{k-1}$, naturally decreases as w_θ increases for $k \geq 2$. This property significantly saves *opportunity cost* during optimization. Specifically, if the policy consistently generates correct responses within k attempts, further reinforcing correct answers becomes unnecessary. Instead, probabilities are more effectively allocated towards exploring alternative tokens, responses, or patterns that promise a higher global payoff.

For Max@k, we also derive a simplified objective without the "hitchhiking" issue. This simplified objective conveys the meaningful concept of *marginal contribution*. Specifically, the gradient weight for a response y reflects the marginal gain in the maximum reward when y is added to sets of $k - 1$ responses.

Based on prior derivations, we can construct unbiased estimators for the Pass@k and Max@k gradient weights. For example, $(1 - w_\theta)^{k-1}$ can be estimated using $\binom{n-c}{k-1} / \binom{n-1}{k-1}$, where n and c denote the number of total and correct samples, respectively.

We conduct a comprehensive evaluation of RSPO on math reasoning. Our results demonstrate that RSPO consistently outperforms the baseline algorithm—which is hindered by the "hitchhiking" problem—across datasets and metrics. Notably, RSPO achieves optimal performance when the training hyperparameter k is aligned with the evaluation metrics Pass@k and Max@k. Finally, we assess

the scalability and robustness of RSPO, confirming its effectiveness under a wide range of conditions.

In summary, this paper makes three key contributions:

- We identify the misalignment problem between standard risk-neutral RL objective and risk-seeking evaluation metrics, which can degrade LLM post-training performance.
- We address the key "hitchhiking" challenge inherent to optimizing Pass@k and Max@k by proposing efficient, unbiased estimators for both metrics and providing rigorous theoretical analyses of their underlying mechanisms.
- We propose the RSPO algorithm, which, through extensive empirical evaluation, effectively maximizes Pass@k and Max@k metrics.

2 Related Literature

Post-training [28] of large language models [15, 34] refers to the refinement stage that follows initial pre-training [35], aiming to adapt a generalized model to specific tasks or to align it with human-defined objectives [2]. Supervised fine-tuning (SFT) [17] is the most direct and widely adopted post-training method. It involves updating model parameters using labeled examples where the desired outputs for specific inputs are explicitly provided. Although straightforward and effective, SFT demands extensive human annotation, significantly limiting its scalability.

To address the limitations of SFT, reinforcement learning (RL) [26] methods have been increasingly utilized in post-training. RL-based post-training seeks to maximize the expected reward, formally represented as $\mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta(y|x)} R(x, y)$. Among RL methods, policy gradient methods have emerged as the dominant framework. Policy gradient [27] algorithms compute the gradient of the RL objective and optimize by sampling according to this gradient, resulting in the loss function $L = -R(x, y) \nabla_\theta \log \pi_\theta(y|x)$. Advanced algorithms such as Trust Region Policy Optimization (TRPO) [18] and Proximal Policy Optimization (PPO) [19] build upon this foundation by imposing constraints on policy updates, thereby ensuring stability and robustness during training. Most recently, Group Relative Policy Optimization (GRPO) [6, 22] has been proposed, wherein multiple responses are sampled, and standard scores of their rewards are employed as gradient weights, thereby eliminating the need for training an additional model for *advantage* estimation.

Despite the accuracy metric (i.e., Pass@1), the evaluation of LLM using verifiable rewards [30] often employs the Pass@k metric. Pass@k, especially when applied to reasoning tasks, is frequently interpreted as reflecting the inherent capacity of an LLM [33]. Notably, the unbiased estimation method for Pass@k introduced by Chen et al. [4] is widely recognized in the LLM literature, although this approach closely aligns with classical U-statistics [10].

Beyond evaluation metrics, selecting the optimal output among n generated candidates—referred to as the best-of-n method [25]—serves as an inference-time strategy. Despite its effectiveness, best-of-n is computationally expensive. Consequently, prior research [5, 20] has primarily focused on distilling a new policy that approximates the best-of-n distribution derived from a fixed, given policy. In contrast, this paper diverges from such approaches by actively searching through an extensive policy space to directly optimize Pass@k or Max@k, rather than approximating a static policy.

3 Preliminary

Notations. Large language models process a prompt x as input and generate a response y as output. The policy π_θ , parameterized by θ , defines the model's behavior: $\pi_\theta(y_t|x, y_{\leq t})$ represents the probability of generating the next token y_t given the prompt x and the previously generated tokens $y_{\leq t}$. Furthermore, $\pi_\theta(y|x)$ denotes the overall probability of generating the complete response y given the prompt x . To evaluate response quality, a reward model $R(x, y)$ assigns a score to the pair (x, y) . This reward can be binary (e.g., indicating correctness) or continuous (e.g., reflecting learned human preferences).

For reading clarity, we introduce a *total order* on the set of responses y by defining $y_1 \leq y_2$ if and only if $R(x, y_1) \leq R(x, y_2)$. Besides, the strict ordering $y_1 < y_2$ implies $R(x, y_1) < R(x, y_2)$.

Problem Formulation. We formally define the Max@k objective for large language model post-training: given an initial policy π_{θ_0} , a dataset of prompts $x \sim D$, a reward model R , the objective is to train a new policy π_θ that maximizes

$$\mathbb{E}_{x \sim \mathcal{D}, y_1, y_2, \dots, y_k \sim \pi_{\theta}(y|x)} [\max\{R(x, y_1), R(x, y_2), \dots, R(x, y_k)\}] \quad (1)$$

Moreover, when rewards are binary ($R(x, y) \in \{0, 1\}$), we fine the Pass@k objective as

$$\mathbb{E}_{x \sim \mathcal{D}, y_1, y_2, \dots, y_k \sim \pi_{\theta}(y|x)} [1 - \prod_{i=1}^k (1 - R(x, y_i))] \quad (2)$$

Baseline. A straightforward approach to optimizing Max@k is to treat the group y_1, y_2, \dots, y_k as a whole and apply policy gradient:

$$\begin{aligned} & \nabla_\theta \mathbb{E}_{x \sim \mathcal{D}, y_1, y_2, \dots, y_k \sim \pi_{\theta}(y|x)} \max_{1 \leq i \leq k} R(x, y_i) \\ &= \nabla_\theta \sum_x P(x) \sum_{y_{1:k}} \pi_\theta(y_{1:k}|x) \max_{1 \leq i \leq k} R(x, y_i) \\ &= \sum_x P(x) \sum_{y_{1:k}} \pi_\theta(y_{1:k}|x) \max_{1 \leq i \leq k} R(x, y_i) \nabla_\theta \log \pi_\theta(y_{1:k}|x) \\ &= \sum_x P(x) \sum_{y_{1:k}} \pi_\theta(y_{1:k}|x) \max_{1 \leq i \leq k} R(x, y_i) \sum_{i=1}^k \nabla_\theta \log \pi_\theta(y_i|x) \\ &\approx \frac{1}{|\mathcal{D}_b|} \sum_{x \sim \mathcal{D}_b} \frac{1}{m} \sum_{j=1}^m \max_{1 \leq i \leq k} R(x, y_i^j) \sum_{i=1}^k \nabla_\theta \log \pi_\theta(y_i^j|x) \end{aligned}$$

The last line employs a Monte Carlo simulation to sample $|\mathcal{D}_b|$ prompts and generate m groups of responses for each prompt.

Within each group, all k responses share the same gradient weight, given by $\max_{1 \leq i \leq k} R(x, y_i^j)$ —the maximum reward among the k responses in that group. This approach introduces what we refer to as the "hitchhiking" problem: low-reward responses may be inadvertently reinforced simply by co-occurring with a high-reward response within the same group. As a result, this can lead to inefficient optimization and suboptimal learning dynamics.

4 Risk Seeking Policy Optimization

4.1 Insight

To address the "hitchhiking" problem, it is essential to decouple individual responses from the sets in which they appear. RSPO addresses this challenge by computing the probability that a given response yields the maximum reward within a set of size k :

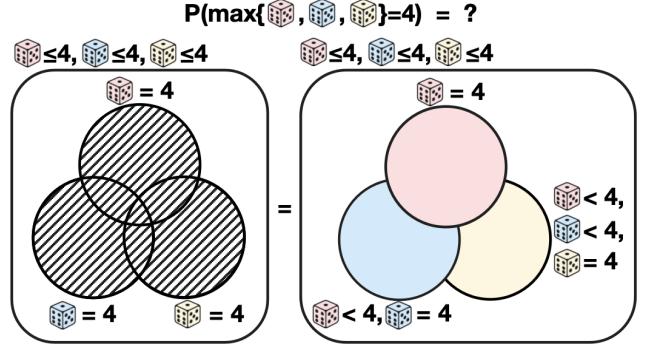


Figure 3: An illustration of how to calculate the probability of achieving the maximum in a group: $P(\max\{D_1, D_2, D_3\} = 4) = P(D_1 = 4, D_2 \leq 4, D_3 \leq 4) + P(D_1 < 4, D_2 = 4, D_3 \leq 4) + P(D_1 < 4, D_2 < 4, D_3 = 4) = \frac{1}{6} \frac{2}{3} \frac{2}{3} + \frac{1}{2} \frac{1}{6} \frac{2}{3} + \frac{1}{2} \frac{1}{2} \frac{1}{6} = \frac{37}{216}$.

THEOREM 4.1. Let y_1, y_2, \dots, y_k be independent samples drawn from the distribution $\pi_\theta(\cdot | x)$. Then the probability that a specific response y is the one with the highest reward among $\{y_i\}_{i=1}^k$ is

$$\begin{aligned} P(y \in \{y_i\}_{i=1}^k, R(x, y) = \max_{1 \leq i \leq k} R(x, y_i)) \\ = \sum_{i=1}^k P_{<, \theta}(y)^{i-1} \cdot \pi_\theta(y|x) \cdot P_{\leq, \theta}(y)^{k-i} \end{aligned} \quad (3)$$

where $P_{<, \theta}(y) = \sum_{y' < y} \pi_\theta(y'|x)$ and $P_{\leq, \theta}(y) = \sum_{y' \leq y} \pi_\theta(y'|x)$.

PROOF. Observe that the event $y \in \{y_i\}_{i=1}^k$ indicates $R(x, y) \leq \max_{1 \leq i \leq k} R(x, y_i)$. This implies:

$$\begin{aligned} & P(y \in \{y_i\}_{i=1}^k, R(x, y) = \max_{1 \leq i \leq k} R(x, y_i)) \\ &= P(y \in \{y_i\}_{i=1}^k, R(x, y) \geq \max_{1 \leq i \leq k} R(x, y_i)) \\ &= P(y \in \{y_i\}_{i=1}^k, R(x, y) \geq R(x, y_i) \forall i) \end{aligned}$$

Moreover, the event $y \in \{y_i\}_{i=1}^k$ can be decomposed as the disjoint union: $P(y \in \{y_i\}_{i=1}^k) = P(y_1 = y) + P(y_1 \neq y, y_2 = y) + \dots + P(y_1 \neq y, y_2 \neq y, \dots, y_{k-1} \neq y, y_k = y)$. As a result,

$$\begin{aligned} & P(y_1 \neq y, y_2 \neq y, \dots, y_{i-1} \neq y, y_i = y, R(x, y) \geq R(x, y_i) \forall i) \\ &= \prod_{j=1}^{i-1} P(R(x, y_j) < R(x, y)) \cdot \pi_\theta(y|x) \cdot \prod_{j=1}^{i-1} P(R(x, y_j) \leq R(x, y)) \\ &= P_{<, \theta}(y)^{i-1} \pi_\theta(y|x) P_{\leq, \theta}(y)^{k-i} \end{aligned}$$

Summing over $i = 1, \dots, k$ therefore yields Theorem 4.1:

$$\begin{aligned} & P(y \in \{y_i\}_{i=1}^k, R(x, y) = \max_{1 \leq i \leq k} R(x, y_i)) \\ &= \sum_{i=1}^k P_{<, \theta}(y)^{i-1} \cdot \pi_\theta(y|x) \cdot P_{\leq, \theta}(y)^{k-i} \end{aligned} \quad \square$$

Figure 3 provides a visual illustration of the proof. It demonstrates how the probability of being the maximum in a group can be computed by decomposing it into a disjoint union of several simpler, computable probabilities.

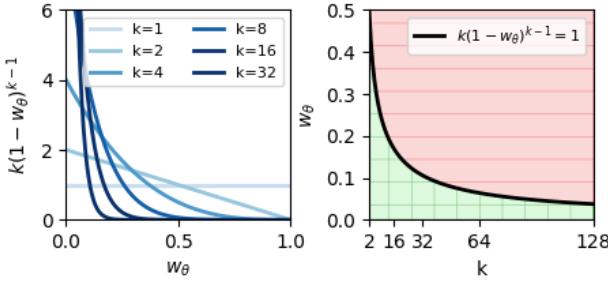


Figure 4: (Left) Weight functions $k(1 - w_\theta)^{k-1}$ as w_θ varies; each curve corresponds to a different k ; (Right) Regions shaded in green indicate parameter pairs (k, w_θ) for which $k(1 - w_\theta)^{k-1} > 1$, while red shading marks $k(1 - w_\theta)^{k-1} < 1$.

By applying Theorem 4.1, we decouple the joint distribution of multiple responses in the Max@ k metric into a distribution over a single response:

THEOREM 4.2. *The original Max@ k objective*

$$\mathbb{E}_{x \sim \mathcal{D}, y_1, y_2, \dots, y_k \sim \pi_\theta(y|x)} [\max\{R(x, y_1), R(x, y_2), \dots, R(x, y_k)\}]$$

is equivalent to

$$\mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta(y|x)} R(x, y) \sum_{i=1}^k P_{<, \theta}(y)^{i-1} P_{\leq, \theta}(y)^{k-i} \quad (4)$$

We further define the probability $w_\theta = \sum_{R(x,y)=1} \pi_\theta(y|x)$ to simplify the expression for the Pass@ k objective:

$$P_{<, \theta}(y) = \begin{cases} 1 - w_\theta & R(x, y) = 1 \\ 0 & R(x, y) = 0 \end{cases}$$

and

$$P_{\leq, \theta}(y) = \begin{cases} 1 & R(x, y) = 1 \\ 1 - w_\theta & R(x, y) = 0 \end{cases}$$

As a result, Theorem 4.1 also yields

COROLLARY 4.2.1. *The original Pass@ k objective*

$$\mathbb{E}_{x \sim \mathcal{D}, y_1, y_2, \dots, y_k \sim \pi_\theta(y|x)} [1 - \prod_{i=1}^k (1 - R(x, y_i))]$$

is equivalent to

$$\mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta(y|x)} R(x, y) \sum_{i=1}^k (1 - w_\theta)^{i-1} \quad (5)$$

4.2 RSPO for Pass@ k

We first study how to optimize the Pass@ k objective by directly applying the policy gradient method:

$$\begin{aligned} & \nabla_\theta \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta(y|x)} R(x, y) \sum_{i=1}^k (1 - w_\theta)^{i-1} \\ & \approx \frac{1}{|\mathcal{D}_b|} \sum_{x \sim \mathcal{D}_b} \frac{1}{m} \sum_{j=1}^m R(x, y_j) \sum_{i=1}^k \nabla_\theta [(1 - w_\theta)^{i-1} \cdot \pi_\theta(y_j|x)] \end{aligned}$$

The difficulty arises in computing $\nabla_\theta [(1 - w_\theta)^{i-1} \cdot \pi_\theta(y^j|x)]$ as it produces a term of $(i-1)\pi_\theta(y^j|x)(1 - w_\theta)^{i-2}\nabla_\theta w_\theta$. This introduces a coupling between the optimization of a particular response y and all other responses y' for which $R(x, y') > 0$. Such nested gradient dependencies across responses are likely a consequence of the underlying "hitchhiking" behavior inherent in the Max@ k objective, and they pose significant challenges for effective optimization.

To address this issue, we observe that although the gradients are interdependent, the additional term $(i-1)\pi_\theta(y^j|x)(1 - w_\theta)^{i-2}\nabla_\theta w_\theta$ consistently involves the factor $\nabla_\theta w_\theta$. This structural pattern suggests that these terms may be globally aggregated or canceled, thereby simplifying the optimization process:

$$\begin{aligned} & \nabla_\theta \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta(y|x)} R(x, y) \sum_{i=1}^k (1 - w_\theta)^{i-1} \\ & = \mathbb{E}_{x \sim \mathcal{D}} \sum_{y \in \mathbf{w}} \nabla_\theta [\pi_\theta(y|x) \sum_{i=1}^k (1 - w_\theta)^{i-1}] \\ & \text{where we denote } \mathbf{w} = \{y | R(x, y) = 1\}. \\ & = \mathbb{E}_{x \sim \mathcal{D}} \sum_{y \in \mathbf{w}} \nabla_\theta [\pi_\theta(y|x) \frac{1 - (1 - w_\theta)^k}{w_\theta}] \end{aligned}$$

because of geometric sum formula.

$$\begin{aligned} & = \mathbb{E}_{x \sim \mathcal{D}} \sum_{y \in \mathbf{w}} \left[\frac{w_\theta \nabla_\theta \pi_\theta - \pi_\theta \nabla_\theta w_\theta}{w_\theta^2} (1 - (1 - w_\theta)^k) \right. \\ & \quad \left. + \frac{\pi_\theta}{w_\theta} \cdot k(1 - w_\theta)^{k-1} \nabla_\theta w_\theta \right] \\ & = \mathbb{E}_{x \sim \mathcal{D}} \left[\frac{w_\theta \sum_{y \in \mathbf{w}} \nabla_\theta \pi_\theta - \sum_{y \in \mathbf{w}} \pi_\theta \nabla_\theta w_\theta}{w_\theta^2} (1 - (1 - w_\theta)^k) \right. \\ & \quad \left. + \frac{\sum_{y \in \mathbf{w}} \pi_\theta}{w_\theta} \cdot k(1 - w_\theta)^{k-1} \nabla_\theta w_\theta \right] \end{aligned}$$

the numerator becomes $w_\theta \nabla_\theta w_\theta - w_\theta \nabla_\theta w_\theta = 0$, yielding

$$\begin{aligned} & \mathbb{E}_{x \sim \mathcal{D}} k(1 - w_\theta)^{k-1} \nabla_\theta w_\theta = \mathbb{E}_{x \sim \mathcal{D}} \sum_{y \in \mathbf{w}} k(1 - w_\theta)^{k-1} \nabla_\theta \pi_\theta(y|x) \\ & = \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta(y|x)} k(1 - w_\theta)^{k-1} R(x, y) \nabla_\theta \log \pi_\theta(y|x) \end{aligned} \quad (6)$$

Surprisingly, despite the underlying complexity of the gradients, the final expression-weighted gradients of the log-likelihood are remarkably clean and interpretable. Specifically, when $R(x, y) = 1$, the weight applied to $\nabla_\theta \log \pi_\theta(y|x)$ is $k(1 - w_\theta)^{k-1}$. Otherwise, when $R(x, y) = 0$, the weight is zero.

We now examine the meaning of the term $k(1 - w_\theta)^{k-1}$. When $k = 1$, the weight simplifies to 1, independent of w_θ , which aligns with the standard reinforcement learning objective—equivalent to a Max@1 formulation. For $k > 1$, the weight $k(1 - w_\theta)^{k-1}$ is a monotonically decreasing function of w_θ , assigning higher importance to smaller values of w_θ and diminishing the influence of larger ones. This behavior becomes more pronounced as k increases.

The left plot in Figure 4 illustrates the decay pattern of $k(1 - w_\theta)^{k-1}$ as w_θ increases. The right plot shows the regime in which RSPO amplifies gradients compared to the conventional RL objective, i.e., when $k(1 - w_\theta)^{k-1} > 1$.

We use the concept of *opportunity cost* to explain why the weight $k(1 - w_\theta)^{k-1}$ decreases with increasing w_θ . Since the total generation probability sums to 1, increasing the probability of one token/response/pattern necessarily decreases the probability assigned to others. Consequently, if the policy is already capable of generating a correct answer within k attempts (e.g., when $w_\theta = 0.5$ and $k = 10$, the probability of generating only incorrect answers is less than $\frac{1}{1000}$), further reinforcement of its win rate becomes redundant. Instead, it is more beneficial to allocate the remaining probability mass to other regions that may yield higher returns.

We then derive estimators for the Pass@k objective, sampling n responses from the policy π_θ for each prompt.

One obvious estimator is

$$\frac{1}{|\mathcal{D}_b|} \sum_{x \sim \mathcal{D}_b} k \left(1 - \frac{1}{n} \sum_{i=1}^n R(x, y^i)\right)^{k-1} \frac{1}{n} \sum_{i=1}^n R(x, y^i) \nabla_\theta \log \pi_\theta(y^i | x) \quad (7)$$

where we estimate $(1 - w_\theta)^{k-1}$ by $(1 - \frac{1}{n} \sum_{i=1}^n R(x, y^i))^{k-1}$.

The advantage of this estimator is that it is both simple and scalable. It leverages n samples to estimate w_θ , independent of the value of k . However, the estimator is biased, as the expectation $E((1 - \frac{1}{n} \sum_{i=1}^n R(x, y^i))^{k-1}) \neq (1 - w_\theta)^{k-1}$.

We then develop an unbiased estimator:

THEOREM 4.3. *When $n \geq k$, an unbiased estimator for Pass@k is:*

$$J_{\text{pass}@k}^{\text{RSPO}}(\theta) = \frac{1}{|\mathcal{D}_b|} \sum_{x \sim \mathcal{D}_b} k \binom{n-c}{k-1} \frac{1}{n} \sum_{i=1}^n R(x, y^i) \nabla_\theta \log \pi_\theta(y^i | x) \quad (8)$$

where $c = \sum_{i=1}^n R(x, y^i)$, i.e., the number of correct responses in the sample.

The objective function $J_{\text{pass}@k}^{\text{RSPO}}(\theta)$ incorporates two key mechanisms to ensure an unbiased gradient estimator. First, when estimating the gradient term $k(1 - w_\theta)^{k-1} R(x, y^i) \nabla_\theta \log \pi_\theta(y^i | x)$, $J_{\text{pass}@k}^{\text{RSPO}}(\theta)$ uses the remaining $n - 1$ responses (excluding y^i) to estimate the weight $(1 - w_\theta)^{k-1}$. This separation guarantees independence between the term $R(x, y^i) \nabla_\theta \log \pi_\theta(y^i | x)$ and the estimator of $(1 - w_\theta)^{k-1}$, thereby ensuring that the expectation of their product equals the product of their expectation ($\mathbb{E}[XY] = \mathbb{E}[X]\mathbb{E}[Y]$). Second, to construct an unbiased estimator of $(1 - w_\theta)^{k-1}$ using the $n - 1$ responses, $J_{\text{pass}@k}^{\text{RSPO}}(\theta)$ enumerates all $\binom{n-1}{k-1}$ subsets of size $k - 1$ and counts the number of subsets, denoted $\binom{n-c}{k-1}$, that consist solely of zero-reward responses. The expectation of each subset matches $(1 - w_\theta)^{k-1}$, yielding an unbiased estimator.

The complete proof of Theorem 4.3 is provided in Appendix B.1.

4.3 RSPO for Max@k

In practice, the Max@k objective often appears alongside a continuous reward model, where the probability $P_{\leq, \theta}(y)$ is approximated by $P_{<, \theta}(y) + \pi(y | x)$. Notably, in large language models, the policy probability $\pi(y | x)$ tends to be extremely small. Given these characteristics, we treat $P_{\leq, \theta}(y)$ and $P_{<, \theta}(y)$ as approximately equal in this section to improve clarity and simplify exposition. For completeness, we provide a rigorous derivation of unbiased estimators without any such approximation in the Appendix C.

Algorithm 1 Risk-Seeking Policy Optimization

Require: Initial policy π_θ , prompts \mathcal{D} , hyperparameters k, n

- 1: **for** step = 1, 2, ... **do**
 - 2: Sample a batch \mathcal{D}_b from \mathcal{D}
 - 3: Sample n responses $\{y^i\}_{i=1}^n \sim \pi_\theta(\cdot | x)$ for each $x \in \mathcal{D}_b$
 - 4: Compute rewards $\{R(x, y^i)\}_{i=1}^n$ for every y^i and x
 - 5: Update policy π_θ by maximizing $J_{\text{pass}@k}^{\text{RSPO}}(\theta)$ or $J_{\text{max}@k}^{\text{RSPO}}(\theta)$
 - 6: **end for**
 - 7: **return** π_θ
-

When we treat $P_{\leq, \theta}(y)$ and $P_{<, \theta}(y)$ as approximately equal, the original Max@k objective

$$\mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta(y|x)} R(x, y) \sum_{i=1}^k P_{<, \theta}(y)^{i-1} P_{\leq, \theta}(y)^{k-i}$$

becomes

$$\mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta(y|x)} R(x, y) \cdot k P_{\leq, \theta}(y)^{k-1} \quad (9)$$

We then develop the gradient of the new objective:

$$\begin{aligned} & \nabla_\theta \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta(y|x)} R(x, y) \cdot k P_{\leq, \theta}(y)^{k-1} \\ &= \mathbb{E}_{x \sim \mathcal{D}} \sum_y R(x, y) k \nabla_\theta [\pi_\theta(y|x) P_{\leq, \theta}(y)^{k-1}] \\ &= \mathbb{E}_{x \sim \mathcal{D}} \sum_y R(x, y) k [P_{\leq, \theta}(y)^{k-1} \nabla_\theta \pi_\theta(y|x) \\ &\quad + (k-1) \pi_\theta(y|x) P_{\leq, \theta}(y)^{k-2} \nabla_\theta P_{\leq, \theta}(y)] \end{aligned}$$

The equations above imply that computing the gradient with respect to y also induces contributions to the gradients of all y' such that $R(x, y') < R(x, y)$. In other words, the gradient of y is influenced not only by its own value but also by all y' for which $R(x, y') > R(x, y)$. Rearranging the terms accordingly yields:

$$\begin{aligned} &= \mathbb{E}_{x \sim \mathcal{D}} \sum_y [k R(x, y) P_{\leq, \theta}(y)^{k-1} \\ &\quad + k(k-1) \sum_{y' \geq y} R(x, y') \pi_\theta(y'|x) P_{\leq, \theta}(y')^{k-2}] \nabla_\theta \pi_\theta(y|x) \\ &= \mathbb{E}_{x \sim \mathcal{D}} \sum_y k R(x, y) P_{\leq, \theta}(y)^{k-1} \nabla_\theta \pi_\theta(y|x) \\ &\quad + \mathbb{E}_{x \sim \mathcal{D}} \sum_y k(k-1) \sum_{y' < y} R(x, y') \pi_\theta(y'|x) P_{\leq, \theta}(y')^{k-2} \nabla_\theta \pi_\theta(y|x) \\ &\quad - \mathbb{E}_{x \sim \mathcal{D}} \sum_y k(k-1) \sum_{y' < y} R(x, y') \pi_\theta(y'|x) P_{\leq, \theta}(y')^{k-2} \nabla_\theta \pi_\theta(y|x) \\ &= \mathbb{E}_{x \sim \mathcal{D}} \sum_y k R(x, y) P_{\leq, \theta}(y)^{k-1} \nabla_\theta \pi_\theta(y|x) \\ &\quad - \mathbb{E}_{x \sim \mathcal{D}} \sum_y k(k-1) \sum_{y' < y} R(x, y') \pi_\theta(y'|x) P_{\leq, \theta}(y')^{k-2} \nabla_\theta \pi_\theta(y|x) \end{aligned}$$

because $\sum_{y'} R(x, y') \pi_\theta(y'|x) P_{\leq, \theta}(y')^{k-2}$ is a constant and $\mathbb{E}_{y \sim \pi_\theta} [c \nabla_\theta \log \pi_\theta(y|x)] = \nabla_\theta c = 0$ for any constant c .

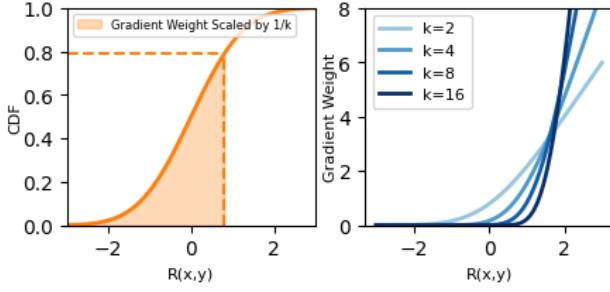


Figure 5: Assume π_θ is a standard Gaussian distribution. (Left) The $k = 2$ case. The curve represents cumulative distribution function of π_θ , corresponding to $\sum_{y' < y} \pi_\theta(y'|x)$. The filled area under the CDF curve thus represents the scaled gradient weight, $\sum_{y' < y} (R(x, y) - R(x, y')) \pi_\theta(y'|x) (k-1) P_{\leq, \theta}(y')^{k-2}$; (Right) Gradient weight $k \sum_{y' < y} (R(x, y) - R(x, y')) \pi_\theta(y'|x) (k-1) P_{\leq, \theta}(y')^{k-2}$ as $R(x, y)$ varies; each curve corresponds to a different k .

As a result, the Max@ k objective is equivalent to

$$\mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta(y|x)} [kR(x, y)P_{\leq, \theta}(y)^{k-1} - k(k-1)g(y)] \nabla_\theta \log \pi_\theta(y|x) \quad (10)$$

where $g(y) = \sum_{y' < y} R(x, y') \pi_\theta(y'|x) P_{\leq, \theta}(y')^{k-2}$.

Although the Max@ k objective is more complex than Pass@ k —primarily due to the involvement of the function $g(y)$, which depends on other responses—we show that it can still be expressed in the form of a weighted sum of log-likelihood gradients. Specifically, the weight associated with each term is given by $kR(x, y)P_{\leq, \theta}(y)^{k-1} - k(k-1)g(y)$.

We show that

$$\begin{aligned} & kR(x, y)P_{\leq, \theta}(y)^{k-1} - k(k-1)g(y) \\ & \approx kR(x, y)P_{\leq, \theta}(y)^{k-1} - k \sum_{y' < y} R(x, y')(k-1)\pi_\theta(y'|x)P_{\leq, \theta}(y')^{k-2} \\ & \approx kR(x, y) \sum_{y' < y} P(y' \in \{y_i\}_{i=1}^{k-1}, R(x, y') = \max_{1 \leq i \leq k-1} R(x, y_i)) \\ & \quad - k \sum_{y' < y} R(x, y')P(y' \in \{y_i\}_{i=1}^{k-1}, R(x, y') = \max_{1 \leq i \leq k-1} R(x, y_i)) \\ & \approx k \sum_{y' < y} (R(x, y) - R(x, y'))\pi_\theta(y'|x)(k-1)P_{\leq, \theta}(y')^{k-2} \end{aligned}$$

The gradient weight can be approximated as $k \sum_{y' < y} (R(x, y) - R(x, y'))\pi_\theta(y'|x)(k-1)P_{\leq, \theta}(y')^{k-2}$, which captures the *marginal contribution* of action y toward increasing the expected maximum reward. This marginal contribution arises from two interacting effects that influence the benefit to the Max@ k objective when the probability mass $\nabla_\theta \pi_\theta(y|x)$ is increased. First, it reflects the higher likelihood that y itself becomes the top-ranked action among the k sampled draws, represented by the $R(x, y)$ term. Second, it accounts for the decreased probability that any lower-reward action $y' < y$ would have otherwise been selected as the maximum, captured by the $R(x, y')$ term. Together, these effects quantify how shifting probability mass toward y improves the expected outcome.

The left plot in Figure 5 depicts the filled area under the cumulative distribution function of $\pi_\theta(y|x)(k-1)P_{\leq, \theta}(y)^{k-2}$, which corresponds to the gradient weight scaled by $\frac{1}{k}$. The right plot illustrates how the gradient weight increases with different k .

The $g(y)$ can be expressed as $\mathbb{E}_{y'} R(x, y') P_{\leq, \theta}(y')^{k-2} \mathbf{1}(R(x, y') < R(x, y))$, which suggests that $g(y)$ can be approximated via Monte Carlo estimation. A straightforward estimator using n responses first estimates $P_{\leq, \theta}(y)$ by $\hat{P}_{\leq, \theta}(y) = \frac{1}{n} \sum_{j=1}^n \mathbf{1}(R(x, y^j) \leq R(x, y))$ and subsequently estimates $g(y)$ by $\hat{g}(y) = \frac{1}{n} \sum_{j=1}^n \mathbf{1}(R(x, y^j) < R(x, y)) R(x, y^j) \hat{P}_{\leq, \theta}(y^j)^{k-2}$, yielding the overall estimator:

$$\frac{k}{n|\mathcal{D}_b|} \sum_{x \sim \mathcal{D}_b} \sum_{i=1}^n [R(x, y^i) \hat{P}_{\leq, \theta}(y^i)^{k-1} - (k-1)\hat{g}(y^i)] \nabla_\theta \log \pi_\theta(y^i|x) \quad (11)$$

Similar to the straightforward estimator used for Pass@ k , this estimator is also simple and scalable, but biased. To address this limitation, we propose an unbiased estimator for Max@ k :

THEOREM 4.4. *When $n \geq k$, an unbiased estimator under mild approximations for Max@ k is:*

$$\begin{aligned} J_{\text{max}@k}^{\text{RSPO}}(\theta) &= \frac{k}{n|\mathcal{D}_b|} \sum_{x \sim \mathcal{D}_b} \sum_{i=1}^n [R(x, y^i) \binom{i-1}{k-1} - \\ &\quad \frac{k-1}{n-1} \sum_{j=1}^{i-1} R(x, y^j) \binom{j-1}{k-2}] \nabla_\theta \log \pi_\theta(y^i|x) \end{aligned} \quad (12)$$

where we sort $\{y^i\}_{i=1}^n$ by $R(x, y_1) < R(x, y_2) < \dots < R(x, y_n)$.

The proof of Theorem 4.4 closely follows Theorem 4.3. When computing the gradient weight for $\nabla_\theta \log \pi_\theta(y^i|x)$, the $J_{\text{max}@k}^{\text{RSPO}}$ utilizes the remaining $n-1$ responses. Furthermore, to estimate $g(y^i)$, it iterates over all $y^j < y^i$. For each fixed pair (i, j) , the $J_{\text{max}@k}^{\text{RSPO}}$ then leverages the remaining $n-2$ responses to estimate $P_{\leq, \theta}(y^j)^{k-2}$ within the computation of $g(y^i)$.

The complete proof of Theorem 4.4 is provided in Appendix B.2.

One key advantage of this estimator lies in the non-negativity of its gradient weights, which remain non-negative regardless of the sign of the reward function R . Importantly, these weights are exactly zero for $i < k$, and strictly positive for $i \geq k$:

$$\begin{aligned} & R(x, y^i) \binom{i-1}{k-1} - \frac{k-1}{n-1} \sum_{j=1}^{i-1} R(x, y^j) \binom{j-1}{k-2} \\ & > R(x, y^i) \left[\binom{i-1}{k-1} - \frac{k-1}{n-1} \sum_{j=1}^{i-1} \binom{j-1}{k-2} \right] \\ & = R(x, y^i) \left[\binom{i-1}{k-1} - \sum_{j=1}^{i-1} \binom{j-1}{k-2} \right] = \frac{R(x, y^i)}{\binom{n-1}{k-1}} \left[\binom{i-1}{k-1} - \sum_{j=k-1}^{i-1} \binom{j-1}{k-2} \right] = 0 \end{aligned}$$

where the last step is due to Hockey-Stick Identity [11].

The non-negativity contributes to training stability, as maximizing the log-likelihood with negative weights can cause gradient explosions due to the convex nature of the logarithmic function. As demonstrated in [1], negative samples necessitate KL constraints to prevent divergence during training, whereas positive samples do not require such regularization.

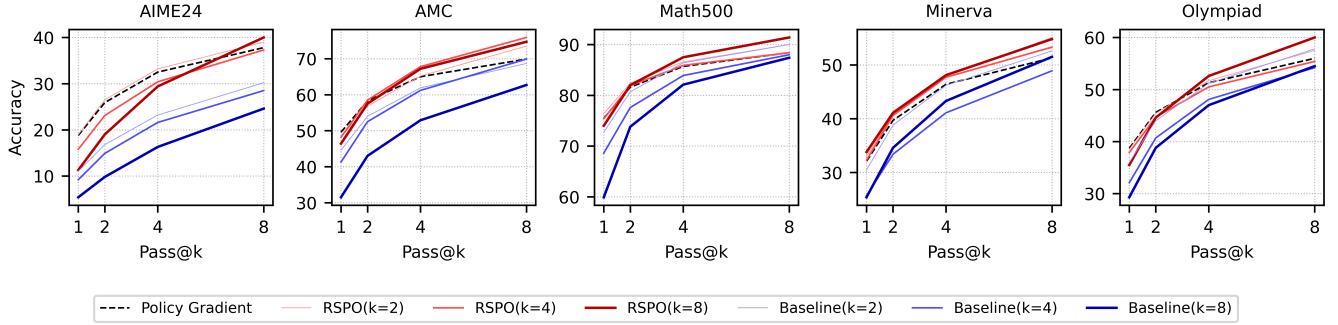


Figure 6: Comparison of model accuracy across five math benchmarks at different Pass@k values.

5 Experiments

5.1 Setup

Following the established experimental setting of large language model post-training, we conduct a comprehensive evaluation on math reasoning. Specifically, we post-train Qwen2.5-Math-1.5B [32] model on Competition Math dataset [8] and assess performance on AIME2024 [13], AMC [13], Math500 [9], Minerva [12], and OlympiadBench [7]. For answer verification, we utilize the xVerify [3]. Our primary focus is on the Pass@k setting, with an additional evaluation of the Max@k setting presented in Section 5.5.

We compare RSPO ($k = 2, 4, 8$) with the baseline algorithm ($k = 2, 4, 8$) mentioned in Section 3 and the policy gradient algorithm, which is equivalent as the special case of RSPO and the baseline when $k = 1$. Unless otherwise specified, all experiments generate 16 responses per prompt ($n = 16$).

To ensure a fair comparison, we maintain identical experimental settings across algorithms. For each training step, we sample 128 prompts from the training set and set the mini-batch size in each step to 256. We repeat the whole training set for 10 epochs. We set the learning rate to $3e - 6$ and set the warm-up ratio to 5%. We conduct our experiments using a server with eight 80GB H800 GPU cards. Each experiment takes approximately 16 hours.

We develop our code based on verl [23] framework. We adhere to the Artifact Pledge and guarantee to release the codes upon publication. Here, we note two useful implementation detail. First, we compute the RSPO gradient weight $\binom{n-c}{k-1} / \binom{n-1}{k-1}$ as $\prod_{i=0}^{k-2} (n-c-i)/(n-1-i)$ instead of computing $\binom{n-c}{k-1}$ and $\binom{n-1}{k-1}$ separately to avoid variable overflow. Second, we remove zero gradient weight responses to reduce training time. This happens when $R(x, y) = 0$ for a response or $n - c < k - 1$ for all responses in a group.

We provide experiment setting justifications in Appendix A.

5.2 Main Analyses

Figure 6 shows the main experiment results. Each graph shows the accuracy achieved at various Pass@k values for a given dataset. Policy Gradient is shown as a black dashed line. RSPO variants are represented by increasingly darker red lines as k increases, while Baseline variants are shown in shades of blue. In addition, thicker lines also indicate larger k values.

First, we observe that across all datasets the blue curves lie consistently below the red curves, demonstrating that RSPO outperforms

the baseline. Moreover, within the baseline family, performance degrades as k increases: the algorithm with $k = 2$ outperforms $k = 4$, which in turn outperforms $k = 8$. This pattern highlights the "hitchhiking" problem inherent to the baselines: low-reward responses are reinforced whenever they co-occur with high-reward responses in the same group. As k grows, this issue intensifies, since low-reward responses have an even greater chance of being boosted alongside their high-reward counterparts.

Second, RSPO tends to achieve its best results when the training hyperparameter k matches the evaluation Pass@k. For instance, on AIME24, the policy-gradient method, i.e. RSPO ($k = 1$), attains its highest accuracy at Pass@1, whereas RSPO trained with $k = 8$ peaks at Pass@8. This trend also holds for other values of k and across all datasets. This finding suggests that practitioners may benefit from choosing their training hyperparameter to coincide with their intended Pass@k evaluation.

Recent work [33] argues that RL may not incentivize reasoning capacity in LLMs, observing that base models outperform their RL-fine-tuned counterparts on Pass@k when k is large. We contend that this phenomenon arises because their training objective ($k = 1$) is misaligned with their evaluation metric ($k > 1$). To resolve this, the community should first agree on the most appropriate choice of k for measuring a model's reasoning capacity, and only then assess whether RL improves reasoning capacity. Our experiments demonstrate that RL—including RSPO—does, in fact, enhance reasoning ability under a suitably chosen Pass@k metric.

Finally, we provide empirical evidence of why RSPO excels on Pass@k. In Section 4.2, we theoretically show that the gradient weight $k(1-w_\theta)^{k-1}$ naturally curbs over-exploitation once Pass@k is sufficiently high. Figure 7 confirms this analysis: as k increases, RSPO maintains higher entropy, yielding more diverse generations and thereby greater opportunities to satisfy the Pass@k criterion.

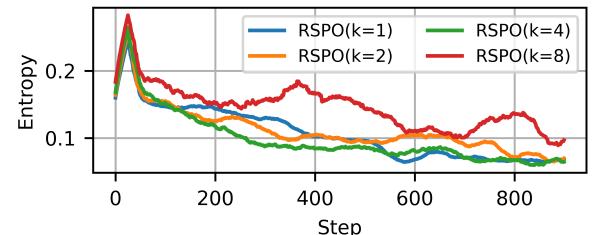


Figure 7: RSPO Entropy over Training Steps.

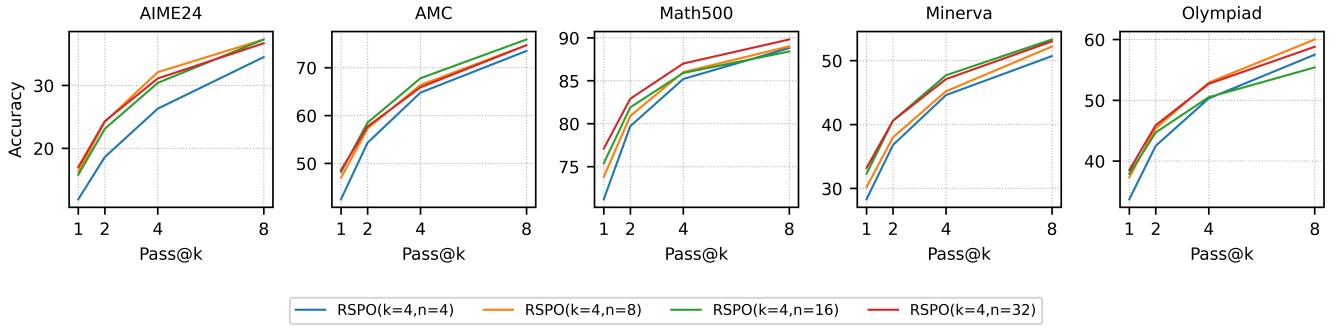
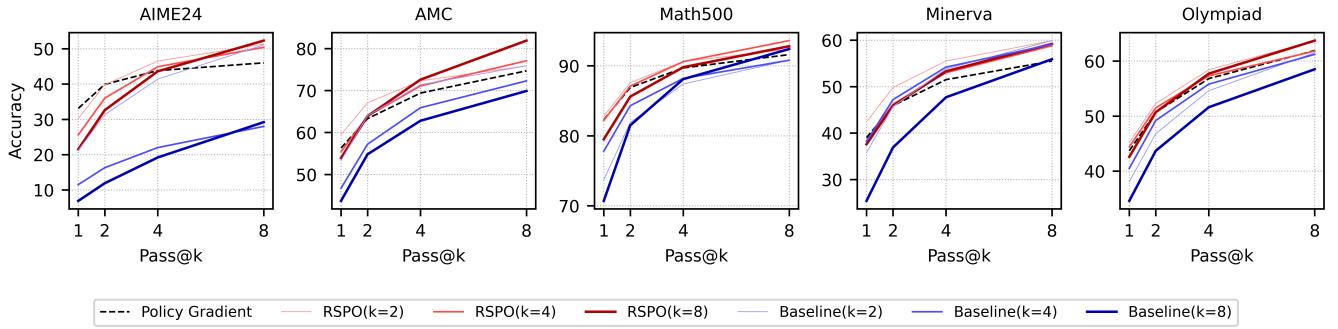
Figure 8: Comparison of model accuracy as n varies at different Pass@k values.

Figure 9: Comparison of model accuracy for the 7B model at different Pass@k values.

5.3 Scalability

In this experiment, we fix $k = 4$ and vary the number of samples per prompt, $n \in \{4, 8, 16, 32\}$, to evaluate the scalability of RSPO. Figure 8 presents the results, showing that increasing n generally improves accuracy across all Pass@k metrics. In particular, $n = 4$ yields the lowest performance regardless of dataset or metric, while $n = 32$ excels across settings. This phenomenon arises because RSPO’s estimator enumerates all $\binom{n-1}{k-1}$ subsets: when $n = k$, only one subset exists, forcing each gradient weight to be either k or 0 with bad flexibility. Consequently, we recommend using at least $n = 2k$, and choosing n as large as computationally feasible.

5.4 Robustness

5.4.1 Alternative Model Size. We further test our algorithm on Qwen2.5-Math-7B, a larger base model than the default setting. Figure 9 shows that the findings in Section 5.2 also hold in the 7B setting, demonstrating the robustness of our algorithm.

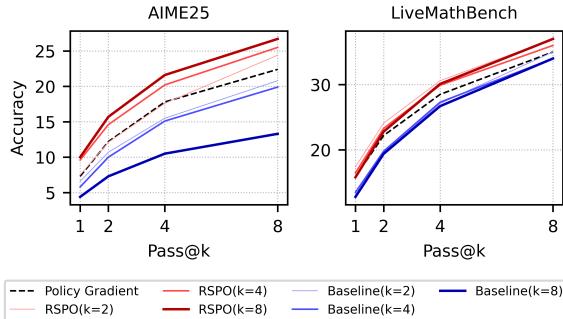


Figure 10: Model accuracy on up-to-date datasets.

5.4.2 Alternative Datasets. Recent study [31] indicates that the AIME2024, AMC, and Math500 datasets were likely included in Qwen2.5’s training data, which could bias experimental outcomes. For example, spurious reward signals may inadvertently boost measured accuracy [21]. To guard against this contamination, we have re-evaluated our results on two additional datasets—AIME2025 [16] and LiveMathBench(202505) [14]—both of which were created after Qwen2.5 and have been verified to be free of overlap with training corpus [31]. As shown in Figure 10, our experimental results remain robust against potential training data contamination.

5.5 Max@k Evaluation

To evaluate the Max@k metric in the context of math reasoning, we design the following reward: $R(x, y) = 1 - 0.5 * \text{len}(y) / \text{max_len}$ if y is correct, otherwise $R(x, y) = 0$. This formulation offers two advantages. First, it avoids the proxy reward problem [24] commonly encountered in RLHF. Second, the reward function is inherently meaningful, as it incentivizes shorter responses, thus lowering inference costs. The Max@k results, presented in Figure 11 in Appendix, follow trends similar to those observed with Pass@k, further validating the effectiveness of RSPO under the Max@k metric.

6 Conclusion

In this paper, we address the mismatch problem between training and evaluation objectives in LLM post-training. We propose RSPO, a method that decouples individual responses from response sets to avoid inefficiencies arising from co-occurrence. We construct unbiased estimators for both Pass@k and Max@k, and support RSPO with clear theoretical analysis and empirical validation.

References

- [1] Abbas Abdolmaleki, Bilal Piot, Bobak Shahriari, Jost Tobias Springenberg, Tim Hertweck, Michael Bloesch, Rishabh Joshi, Thomas Lampe, Junhyuk Oh, Nicolas Heess, et al. [n. d.]. Learning from negative feedback, or positive feedback or both. In *The Thirteenth International Conference on Learning Representations*.
- [2] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862* (2022).
- [3] Ding Chen, Qingchen Yu, Pengyuan Wang, Wentao Zhang, Bo Tang, Feiyu Xiong, Xinch Li, Minchuan Yang, and Zhiyi Li. 2025. xVerify: Efficient Answer Verifier for Reasoning Model Evaluations. *arXiv:2504.10481* [cs.CL] <https://arxiv.org/abs/2504.10481>
- [4] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374* (2021).
- [5] Lin Gui, Cristina Gârbacean, and Victor Veitch. 2024. Bonbon alignment for large language models and the sweetness of best-of-n sampling. *Advances in Neural Information Processing Systems 37* (2024), 2851–2885.
- [6] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948* (2025).
- [7] Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. 2024. Olympiad-bench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008* (2024).
- [8] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring Mathematical Problem Solving With the MATH Dataset. *arXiv preprint arXiv:2103.03874* (2021).
- [9] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874* (2021).
- [10] Wassily Hoeffding. 1992. A class of statistics with asymptotically normal distribution. *Breakthroughs in statistics: Foundations and basic theory* (1992), 308–334.
- [11] Charles H Jones. 1996. Generalized hockey stick identities and N-dimensional blockwalking. *The Fibonacci Quarterly* 34, 3 (1996), 280–288.
- [12] Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Sloane, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. 2022. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems 35* (2022), 3843–3857.
- [13] Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Zijiu Shen, et al. 2024. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository* 13 (2024), 9.
- [14] Junnan Liu, Hongwei Liu, Linchen Xiao, Ziyi Wang, Kuikun Liu, Songyang Gao, Wenwei Zhang, Songyang Zhang, and Kai Chen. 2024. Are Your LLMs Capable of Stable Reasoning? *arXiv preprint arXiv:2412.13147* (2024).
- [15] Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2025. Large Language Models: A Survey. *arXiv:2402.06196* [cs.CL] <https://arxiv.org/abs/2402.06196>
- [16] opencompass. 2025. AIME2025: Problems from the American Invitational Mathematics Examination (2025) – I & II. <https://huggingface.co/datasets/opencompass/AIME2025>. Dataset, accessed 22 July 2025.
- [17] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems 35* (2022), 27730–27744.
- [18] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. 2015. Trust Region Policy Optimization. In *Proceedings of the 32nd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 37)*, Francis Bach and David Blei (Eds.). PMLR, Lille, France, 1889–1897. <https://proceedings.mlr.press/v37/schulman15.html>
- [19] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).
- [20] Pier Giuseppe Sessa, Robert Dadashi, Léonard Hussenot, Johan Ferret, Nino Vieillard, Alexandre Ramé, Bobak Shahriari, Sarah Perrin, Abe Friesen, Geoffrey Cideron, et al. 2024. Bond: Aligning llms with best-of-n distillation. *arXiv preprint arXiv:2407.14622* (2024).
- [21] Rulin Shao, Shuyue Stella Li, Rui Xin, Scott Geng, Yiping Wang, Sewoong Oh, Simon Shaolei Du, Nathan Lambert, Sewon Min, Ranjay Krishna, et al. 2025. Spurious rewards: Rethinking training signals in rlrv. *arXiv preprint arXiv:2506.10947* (2025).
- [22] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300* (2024).
- [23] Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2024. HybridFlow: A Flexible and Efficient RLHF Framework. *arXiv preprint arXiv: 2409.19256* (2024).
- [24] Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. 2022. Defining and characterizing reward gaming. *Advances in Neural Information Processing Systems 35* (2022), 9460–9471.
- [25] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in neural information processing systems 33* (2020), 3008–3021.
- [26] Richard S Sutton, Andrew G Barto, et al. 1998. *Reinforcement learning: An introduction*. Vol. 1. MIT press Cambridge.
- [27] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems 12* (1999).
- [28] Guiyao Tie, Zeli Zhao, Dingjie Song, Fuyang Wei, Rong Zhou, Yurou Dai, Wen Yin, Zhejian Yang, Jianguo Yan, Yao Su, Zhenhan Dai, Yifeng Xie, Yihan Cao, Lichao Sun, Pan Zhou, Lifang He, Hechang Chen, Yu Zhang, Qingsong Wen, Tianming Liu, Neil Zhenqiang Gong, Jiliang Tang, Caiming Xiong, Heng Ji, Philip S. Yu, and Jianfeng Gao. 2025. A Survey on Post-training of Large Language Models. *arXiv:2503.06072* [cs.CL] <https://arxiv.org/abs/2503.06072>
- [29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems 30* (2017).
- [30] Peng-Yuan Wang, Tian-Shuo Liu, Chenyang Wang, Yi-Di Wang, Shu Yan, Cheng-Xing Jia, Xu-Hui Liu, Xin-Wei Chen, Jia-Cheng Xu, Ziniu Li, et al. 2025. A Survey on Large Language Models for Mathematical Reasoning. *arXiv preprint arXiv:2506.08446* (2025).
- [31] Mingqi Wu, Zhihao Zhang, Qiaole Dong, Zhiheng Xi, Jun Zhao, Senjie Jin, Xiaoran Fan, Yuhao Zhou, Yanwei Fu, Qin Liu, et al. 2025. Reasoning or Memorization? Unreliable Results of Reinforcement Learning Due to Data Contamination. *arXiv preprint arXiv:2507.10532* (2025).
- [32] 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 arXiv:2409.12122* (2024).
- [33] 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 arXiv:2504.13837* (2025).
- [34] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2025. A Survey of Large Language Models. *arXiv:2303.18223* [cs.CL] <https://arxiv.org/abs/2303.18223>
- [35] Ce Zhou, Qian Li, Chen Li, Jun Yu, Yixin Liu, Guangjing Wang, Kai Zhang, Cheng Ji, Qiben Yan, Lifang He, Hao Peng, Jianxin Li, Jia Wu, Ziwei Liu, Pengtao Xie, Caiming Xiong, Jian Pei, Philip S. Yu, and Lichao Sun. 2023. A Comprehensive Survey on Pretrained Foundation Models: A History from BERT to ChatGPT. *arXiv:2302.09419* [cs.AI] <https://arxiv.org/abs/2302.09419>

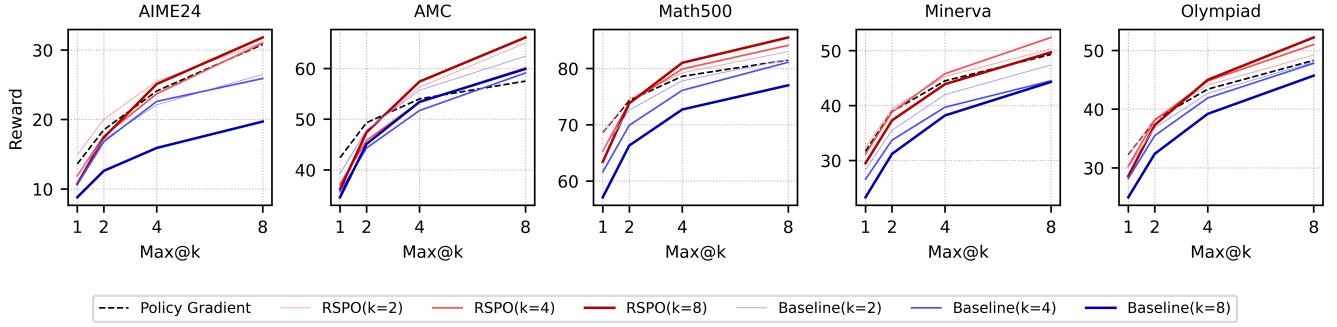


Figure 11: Model comparison across five math benchmarks at different Max@k values.

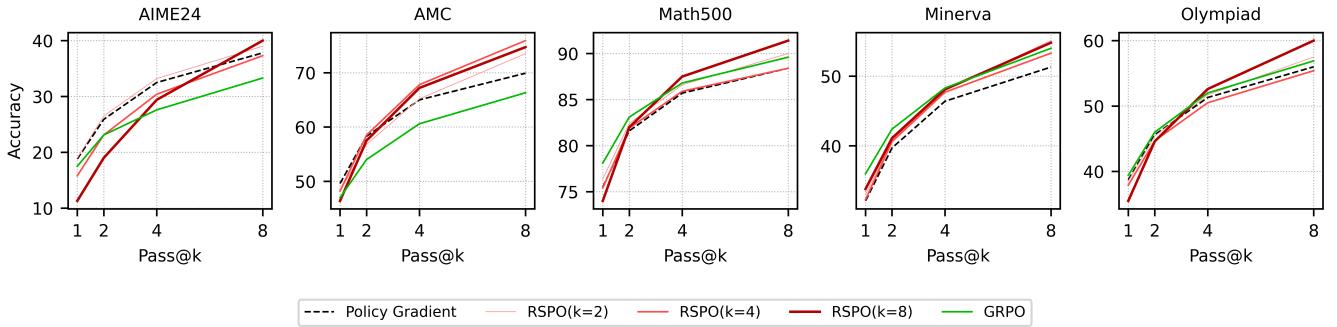


Figure 12: Model comparison with GRPO across five math benchmarks at different Pass@k values.

Appendix

A Experiment Setting Justifications

In this appendix, we provide a justification for our experimental setup and address potential concerns that reviewers may raise.

1. Why are the experiment figures dense? Why not use tables instead?

There are two reasons for this choice. First, tables would occupy significantly more space than figures. A table would likely span an entire page, whereas a figure is more compact. Second, detecting trends in tables can be challenging. For example, the relationship between different values of k may be difficult to discern. In contrast, figures provide a clearer visual representation of these trends.

2. Why don't you compare RSPO with other popular RL algorithms?

We justify this experimental setup both theoretically and empirically. From a theoretical perspective, existing RL algorithms for LLM post-training generally optimize the risk-neutral objective $\mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta(y|x)} R(x, y)$. In contrast, our focus is on the Pass@k and Max@k metrics, where the objective mismatch would lead to suboptimal performance of the existing RL algorithms in evaluation.

Empirically, we tested the performance of the widely-used GRPO [22] in Figure 12 and observed that it exhibits performance and trends similar to those of the Policy Gradient method. This finding suggests that Policy Gradient serves as a reasonable representative baseline for algorithms that optimize the risk-neutral objective.

3. Why don't you test Max@k on alignment/RLHF tasks?

LLM post-training tasks generally fall into two categories: reinforcement learning with verifiable rewards (RLVR) and reinforcement learning from human feedback (RLHF). In the RLHF setting,

the effectiveness of RL algorithms depends on how accurately the proxy rewards align with the true rewards. This *proxy reward issue* [24] is a well-documented challenge in reinforcement learning. As a result, it is common practice [22, 31, 33] to evaluate post-training algorithms in the RLVR setting, where rewards are verifiable and more directly related to performance.

B Proofs

B.1 Proof of Theorem 4.3

PROOF. Consider using y^i to estimate

$$\mathbb{E}_{y \sim \pi_\theta(y|x)} R(x, y) \nabla_\theta \log \pi_\theta(y|x)$$

and $\mathbf{y}^{-i} = \{y^j | 1 \leq j \leq n, j \neq i\}$ to estimate $(1 - w_\theta)^{k-1}$.

Obviously, $R(x, y^i) \nabla_\theta \log \pi_\theta(y^i|x)$ is an unbiased estimator of

$$\mathbb{E}_{y \sim \pi_\theta(y|x)} R(x, y) \nabla_\theta \log \pi_\theta(y|x), \text{ because}$$

$$\begin{aligned} & \mathbb{E}_{y^i \sim \pi_\theta(y|x)} R(x, y^i) \nabla_\theta \log \pi_\theta(y^i|x) \\ &= \mathbb{E}_{y \sim \pi_\theta(y|x)} R(x, y) \nabla_\theta \log \pi_\theta(y|x) \end{aligned} \tag{13}$$

Next we prove that $\frac{\binom{n-1-c^{-i}}{k-1}}{\binom{n-1}{k-1}}$, with $c^{-i} = \sum_{y^j \in \mathbf{y}^{-i}} R(x, y^j)$, is an unbiased estimator of $(1 - w_\theta)^{k-1}$:

For a size $k-1$ subset s of \mathbf{y}^{-i} ($|s| = k-1, s \subseteq \mathbf{y}^{-i}$), we denote

$$I(s) = \begin{cases} 1 & R(x, y^j) = 0, \forall y^j \in s \\ 0 & \text{otherwise} \end{cases}$$

We have

$$\mathbb{E}(I(s)) = \mathbb{E}\left(\prod_{y^j \in s} (1 - R(x, y^j))^{k-1}\right) = (1 - w_\theta)^{k-1} \quad (14)$$

Then

$$\begin{aligned} \mathbb{E}\left(\frac{\binom{n-1-c}{k-1}}{\binom{n-1}{k-1}}\right) &= \frac{1}{\binom{n-1}{k-1}} \mathbb{E}\left(\sum_{|s|=k-1, s \subseteq \mathbf{y}^{-i}} I(s)\right) \\ &= \frac{1}{\binom{n-1}{k-1}} \sum_{|s|=k-1, s \subseteq \mathbf{y}^{-i}} \mathbb{E}(I(s)) = \frac{1}{\binom{n-1}{k-1}} \sum_{|s|=k-1, s \subseteq \mathbf{y}^{-i}} (1 - w_\theta)^{k-1} \\ &= \frac{1}{\binom{n-1}{k-1}} \binom{n-1}{k-1} (1 - w_\theta)^{k-1} = (1 - w_\theta)^{k-1} \end{aligned} \quad (15)$$

As y^i are independent of \mathbf{y}^{-i} ,

$$\begin{aligned} &\mathbb{E}\left(\frac{\binom{n-1-c}{k-1}}{\binom{n-1}{k-1}} \cdot R(x, y^i) \nabla_\theta \log \pi_\theta(y^i | x)\right) \\ &= \mathbb{E}\left(\frac{\binom{n-1-c}{k-1}}{\binom{n-1}{k-1}}\right) \cdot \mathbb{E}(R(x, y^i) \nabla_\theta \log \pi_\theta(y^i | x)) \\ &= (1 - w_\theta)^{k-1} \mathbb{E}(R(x, y^i) \nabla_\theta \log \pi_\theta(y^i | x)) \\ &= \mathbb{E}((1 - w_\theta)^{k-1} R(x, y^i) \nabla_\theta \log \pi_\theta(y^i | x)) \end{aligned} \quad (16)$$

As a result, $\frac{\binom{n-1-c}{k-1}}{\binom{n-1}{k-1}} \cdot R(x, y^i) \nabla_\theta \log \pi_\theta(y^i | x)$ is an unbiased estimator of original Pass@k objective.

Because the above proof holds for every i ,

$$\begin{aligned} &\frac{1}{n} \sum_{i=1}^n \frac{\binom{n-1-c}{k-1}}{\binom{n-1}{k-1}} \cdot R(x, y^i) \nabla_\theta \log \pi_\theta(y^i | x) \\ &= \frac{(n-c)}{\binom{n-1}{k-1}} \frac{1}{n} \sum_{i=1}^n R(x, y^i) \nabla_\theta \log \pi_\theta(y^i | x) \end{aligned} \quad (17)$$

is also an unbiased estimator. The theorem has been proofed. \square

B.2 Proof of Theorem 4.4

PROOF. For a subset s of $\{y^i\}_{i=1}^n$, we denote

$$I(s|y) = \begin{cases} 1 & R(x, y') \leq R(x, y), \forall y' \in s \\ 0 & \text{otherwise} \end{cases}$$

We first prove:

$$\begin{aligned} \mathbb{E}\left[\frac{\binom{i-1}{k-1}}{\binom{n-1}{k-1}} | y^i\right] &= \mathbb{E}\left[\frac{1}{\binom{n-1}{k-1}} \sum_{\substack{|s|=k-1, \\ s \subseteq \{y^j | 1 \leq j \leq n, j \neq i\}}} I(s|y^i) | y^i\right] \\ &= \frac{1}{\binom{n-1}{k-1}} \sum_{\substack{|s|=k-1, \\ s \subseteq \{y^j | 1 \leq j \leq n, j \neq i\}}} \mathbb{E}[I(s|y^i) | y^i] \\ &= \frac{1}{\binom{n-1}{k-1}} \sum_{\substack{|s|=k-1, \\ s \subseteq \{y^j | 1 \leq j \leq n, j \neq i\}}} P_{\leq, \theta}(y^i)^{k-1} \\ &= \frac{\binom{n-1}{k-1}}{\binom{n-1}{k-1}} P_{\leq, \theta}(y^i)^{k-1} = P_{\leq, \theta}(y^i)^{k-1} \end{aligned} \quad (18)$$

Multiplying $R(x, y^i) \nabla_\theta \log \pi_\theta(y^i | x)$ to both sides,

$$\begin{aligned} &R(x, y^i) P_{\leq, \theta}(y^i)^{k-1} \nabla_\theta \log \pi_\theta(y^i | x) \\ &= R(x, y^i) \mathbb{E}\left[\frac{\binom{i-1}{k-1}}{\binom{n-1}{k-1}} | y^i\right] \nabla_\theta \log \pi_\theta(y^i | x) \\ &= \mathbb{E}[R(x, y^i) \frac{\binom{i-1}{k-1}}{\binom{n-1}{k-1}} \nabla_\theta \log \pi_\theta(y^i | x) | y^i] \end{aligned} \quad (19)$$

Taking expectation to both sides yield:

$$\begin{aligned} &\mathbb{E}[R(x, y) P_{\leq, \theta}(y)^{k-1} \nabla_\theta \log \pi_\theta(y | x)] \\ &= \mathbb{E}[R(x, y^i) P_{\leq, \theta}(y^i)^{k-1} \nabla_\theta \log \pi_\theta(y^i | x)] \\ &= \mathbb{E}[\mathbb{E}[R(x, y^i) \frac{\binom{i-1}{k-1}}{\binom{n-1}{k-1}} \nabla_\theta \log \pi_\theta(y^i | x) | y^i]] \\ &= \mathbb{E}[R(x, y^i) \frac{\binom{i-1}{k-1}}{\binom{n-1}{k-1}} \nabla_\theta \log \pi_\theta(y^i | x)] \end{aligned} \quad (20)$$

Because the above proof holds for every i ,

$$\begin{aligned} &\mathbb{E}\left[\frac{1}{n} \sum_{i=1}^n R(x, y^i) \frac{\binom{i-1}{k-1}}{\binom{n-1}{k-1}} \nabla_\theta \log \pi_\theta(y^i | x)\right] \\ &= \frac{1}{n} \sum_{i=1}^n \mathbb{E}[R(x, y^i) \frac{\binom{i-1}{k-1}}{\binom{n-1}{k-1}} \nabla_\theta \log \pi_\theta(y^i | x)] \\ &= \mathbb{E}[R(x, y) P_{\leq, \theta}(y)^{k-1} \nabla_\theta \log \pi_\theta(y | x)] \end{aligned} \quad (21)$$

Similarly, we can proof

$$\begin{aligned} &\mathbb{E}\left[\frac{1}{n-1} \sum_{j=1}^{i-1} R(x, y^j) \frac{\binom{j-1}{k-2}}{\binom{n-2}{k-2}} | y^i\right] \\ &= \mathbb{E}\left[\frac{1}{n-1} \left[\sum_{j=1}^{i-1} R(x, y^j) \frac{\binom{j-1}{k-2}}{\binom{n-2}{k-2}} + \sum_{j=i+1}^n 0 \right] | y^i\right] \\ &= \mathbb{E}\left[\sum_{y'} \pi_\theta(y' | x) R(x, y') P_{\leq, \theta}(y')^{k-2} \mathbf{1}(R(x, y') < R(x, y^i)) | y^i\right] \\ &= \mathbb{E}\left[\sum_{y' < y^i} R(x, y') \pi_\theta(y' | x) P_{\leq, \theta}(y')^{k-2} | y^i\right] = \mathbb{E}[g(y^i) | y^i] \end{aligned} \quad (22)$$

in which $\mathbb{E}\left[\frac{\binom{j-1}{k-2}}{\binom{n-2}{k-2}} | y^i, y^j\right] = \mathbb{E}[P_{\leq, \theta}(y^j)^{k-2} | y^i]$.

As a result,

$$\begin{aligned} &\mathbb{E}\left[\frac{1}{n} \sum_{i=1}^n \frac{k-1}{n-1} \sum_{j=1}^{i-1} R(x, y^j) \frac{\binom{j-1}{k-2}}{\binom{n-2}{k-2}} \nabla_\theta \log \pi_\theta(y^i | x)\right] \\ &= \mathbb{E}[(k-1)g(y) \nabla_\theta \log \pi_\theta(y | x)] \end{aligned} \quad (23)$$

Since both parts are unbiased, the theorem has been proofed. \square

C RSPO for Max@k without approximation

In Section 4.3, we approximated $P_{\leq,\theta}(y)$ and $P_{<,\theta}(y)$ equally for simplicity and clearance. This approximation is suitable for reward models which generate continues rewards. However, when reward models generate discrete rewards—for example, generative rewards models—such approximation may lead to significant misalignment.

In this section, we develop unbiased estimators for Max@k without any approximation. We start from the original Max@k object:

$$\begin{aligned} & \nabla_{\theta} \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_{\theta}(y|x)} R(x, y) \sum_{t=1}^k P_{<,\theta}(y)^{t-1} P_{\leq,\theta}(y)^{k-t} \\ &= \mathbb{E}_{x \sim \mathcal{D}} \sum_{t=1}^k \sum_y R(x, y) \nabla_{\theta} [\pi_{\theta}(y|x) P_{<,\theta}(y)^{t-1} P_{\leq,\theta}(y)^{k-t}] \\ &= \mathbb{E}_{x \sim \mathcal{D}} \sum_{t=1}^k \sum_y R(x, y) P_{<,\theta}(y)^{t-1} P_{\leq,\theta}(y)^{k-t} \nabla_{\theta} \pi_{\theta}(y|x) \\ &+ \mathbb{E}_{x \sim \mathcal{D}} \sum_{t=1}^k \sum_y R(x, y) \pi_{\theta}(y|x) (t-1) P_{<,\theta}(y)^{t-2} P_{\leq,\theta}(y)^{k-t} \nabla_{\theta} P_{<,\theta}(y) \\ &+ \mathbb{E}_{x \sim \mathcal{D}} \sum_{t=1}^k \sum_y R(x, y) \pi_{\theta}(y|x) P_{<,\theta}(y)^{t-1} (k-t) P_{\leq,\theta}(y)^{k-t-1} \nabla_{\theta} P_{\leq,\theta}(y) \end{aligned}$$

gather terms related to $\nabla_{\theta} \pi_{\theta}(y|x)$ together :

$$\begin{aligned} &= \mathbb{E}_{x \sim \mathcal{D}} \sum_{t=1}^k \sum_y [R(x, y) P_{<,\theta}(y)^{t-1} P_{\leq,\theta}(y)^{k-t} \\ &+ \sum_{y' > y} R(x, y') \pi_{\theta}(y'|x) (t-1) P_{<,\theta}(y')^{t-2} P_{\leq,\theta}(y')^{k-t} \\ &+ \sum_{y' \geq y} R(x, y') \pi_{\theta}(y'|x) P_{<,\theta}(y')^{t-1} (k-t) P_{\leq,\theta}(y')^{k-t-1}] \nabla_{\theta} \pi_{\theta}(y|x) \end{aligned}$$

because $\mathbb{E}_{y \sim \pi_{\theta}} [c \nabla_{\theta} \log \pi_{\theta}(y|x)] = \nabla_{\theta} c = 0$ for any constant c :

$$\begin{aligned} &= \mathbb{E}_{x \sim \mathcal{D}} \sum_{t=1}^k \sum_y [R(x, y) P_{<,\theta}(y)^{t-1} P_{\leq,\theta}(y)^{k-t} \\ &- \sum_{y' \leq y} R(x, y') \pi_{\theta}(y'|x) (t-1) P_{<,\theta}(y')^{t-2} P_{\leq,\theta}(y')^{k-t} \\ &- \sum_{y' < y} R(x, y') \pi_{\theta}(y'|x) P_{<,\theta}(y')^{t-1} (k-t) P_{\leq,\theta}(y')^{k-t-1}] \nabla_{\theta} \pi_{\theta}(y|x) \end{aligned}$$

rearrange terms to remove $t-1, k-t$ for $y' < y$:

$$\begin{aligned} &= \mathbb{E}_{x \sim \mathcal{D}} \sum_y [R(x, y) \sum_{t=1}^k P_{<,\theta}(y)^{t-1} P_{\leq,\theta}(y)^{k-t} \\ &- \sum_{y'=y} R(x, y') \pi_{\theta}(y'|x) \sum_{t=1}^{k-1} t P_{<,\theta}(y')^{t-1} P_{\leq,\theta}(y')^{k-t-1} \\ &- \sum_{y' < y} R(x, y') \pi_{\theta}(y'|x) \sum_{t=1}^{k-1} k P_{<,\theta}(y')^{t-1} P_{\leq,\theta}(y')^{k-t-1}] \nabla_{\theta} \pi_{\theta}(y|x) \end{aligned}$$

Note that $y' = y$ indicates $R(x, y') = R(x, y)$ according to our definition. In other words, the second term iterate all the responses with the equivalent reward as y .

Similar as Section 4.3, we will use $n-1$ responses excluding y to estimate $P_{<,\theta}(y)^{t-1} P_{\leq,\theta}(y)^{k-t}$ and $n-2$ responses excluding y, y' to estimate $P_{<,\theta}(y')^{t-1} P_{\leq,\theta}(y')^{k-t-1}$. We show that

LEMMA C.1. *Given n_0 sampled responses, in which $c_<$ responses have less rewards than y_0 and $c_=$ responses have equivalent rewards as y_0 , an unbiased estimator of $P_{<,\theta}(y_0)^a P_{\leq,\theta}(y_0)^b$ is $\frac{\binom{c_<}{a} \binom{c_< + c_= - a}{b}}{\binom{n_0}{a+b}}$ when $n_0 \geq a+b$.*

Using this lemma, we are able to compute an unbiased estimator for the Max@k object by Algorithm 2.

Algorithm 2 An Unbiased Estimator for Max@k

Require: prompt x , number of samples n , Max@k size k , policy π_{θ} , reward function R

1: Sample n responses from the policy:

```

1:    $\{y_1, \dots, y_n\} \stackrel{\text{iid}}{\sim} \pi_{\theta}(\cdot | x)$ 
2:   initialize gradient estimator  $\hat{g} \leftarrow 0$ 
3:   for  $i = 1, \dots, n$  do                                ▶ Loop over main sample  $y_i$ 
4:      $y \leftarrow y_i, r \leftarrow R(x, y)$ 
5:     let
6:        $c_< \leftarrow |\{j \neq i : R(x, y_j) < r\}|, c_= \leftarrow |\{j \neq i : R(x, y_j) = r\}|$ 
7:       for  $t = 1, \dots, k$  do                          ▶ First term estimation
8:          $\hat{E}_{i,t}^{(1)} = \frac{\binom{c_<}{t-1} \binom{c_< + c_= - (t-1)}{k-t}}{\binom{n-1}{k-1} \binom{k-1}{t-1}}$ 
9:          $\hat{g} += r \hat{E}_{i,t}^{(1)} \nabla_{\theta} \log \pi_{\theta}(y | x)$ 
10:        end for
11:        for each  $j \neq i$  do                      ▶ Second and third term estimation
12:           $y' \leftarrow y_j, r' \leftarrow R(x, y')$ 
13:          let
14:             $c'_< \leftarrow |\{\ell \neq i, j : R(x, y_{\ell}) < r'\}|, c'_= \leftarrow |\{\ell \neq i, j : R(x, y_{\ell}) = r'\}|$ 
15:            for  $t = 1, \dots, k-1$  do
16:               $\hat{E}_{ij,t}^{(2,3)} = \frac{\binom{c'_<}{t-1} \binom{c'_< + c'_= - (t-1)}{k-t-1}}{\binom{n-2}{k-2} \binom{k-2}{t-1}}$ 
17:               $\hat{g} -= r' \pi_{\theta}(y' | x) \frac{t}{n-1} \hat{E}_{ij,t}^{(2,3)} \nabla_{\theta} \log \pi_{\theta}(y | x)$ 
18:            else if  $r' < r$  then                      ▶ Third term
19:               $\hat{g} -= r' \pi_{\theta}(y' | x) \frac{k}{n-1} \hat{E}_{ij,t}^{(2,3)} \nabla_{\theta} \log \pi_{\theta}(y | x)$ 
20:            end if
21:          end for
22:        end for
23:      end for
24:    return  $\hat{g}/n$ 

```

We then derive a closed-form unbiased estimator without summing over t , which simplifies computation and enables comparisons with other estimators.

First, we show that

LEMMA C.2.

$$\sum_{0 \leq a \leq m, b=m-a} \frac{\binom{c_<}{a} \binom{c_<+c_= - a}{b}}{\binom{a+b}{a}} = \frac{m+1}{c_= + 1} [\binom{c_<+c_= + 1}{m+1} - \binom{c_<}{m+1}] \quad (24)$$

Using this lemma, we can derive an unbiased estimator

$$\frac{1}{\binom{n-2}{k-2}} \frac{k-1}{c_= + 1} [\binom{c'_<+c'_= + 1}{k-1} - \binom{c'_<}{k-1}] \quad (25)$$

for $\sum_{t=1}^{k-1} P_{<,\theta}(y')^{t-1} P_{\leq,\theta}(y')^{k-t-1}$. The $c'_<$ and $c'_=$ are the number of responses with rewards less than or equal to y' among the $n-2$ responses, respectively.

Moreover, as the rest of $c'_=$ responses have the equivalent reward as y' , they can form a group of size $c'_= + 1$, in which each of them has the same weight. As a result, the total contribution of this group to the weight of $\nabla_\theta \pi_\theta(y|x)$ is

$$\begin{aligned} & -\frac{k}{n-1} \frac{c'_= + 1}{\binom{n-2}{k-2}} \frac{k-1}{c'_= + 1} [\binom{c'_<+c'_= + 1}{k-1} - \binom{c'_<}{k-1}] R(x, y') \\ & = -\frac{1}{\binom{n-2}{k-2}} \frac{k(k-1)}{n-1} [\binom{c'_<+c'_= + 1}{k-1} - \binom{c'_<}{k-1}] R(x, y') \\ & = -\frac{1}{\binom{n-2}{k-2}} \frac{k(k-1)}{n-1} [\binom{c'_<}{k-2} + \binom{c'_<+1}{k-2} + \dots + \binom{c'_<+c'_=}{k-2}] R(x, y') \end{aligned} \quad (26)$$

the estimator for the third term ($y' < y$) is the sum of every group. The last step ensures when this group of responses are sorted into non-decreasing order from index $c'_< + 1$ to index $c'_< + c'_= + 1$, the formula can be expressed using their index -1 .

Next, we compute the estimator for the first (y) and second term ($y' = y$). Using Lemma C.2, we can also compute the contribution of first term to the weight of $\nabla_\theta \pi_\theta(y|x)$:

$$\frac{1}{\binom{n-1}{k-1}} \frac{k}{c_= + 1} [\binom{c_<+c_= + 1}{k} - \binom{c_<}{k}] R(x, y) \quad (27)$$

for $\sum_{t=1}^k P_{<,\theta}(y)^{t-1} P_{\leq,\theta}(y)^{k-t}$ in the first term. The $c_<$ and $c_=$ are the number of responses with rewards less than or equal to y among the $n-1$ responses, respectively.

For the second term, we show that

LEMMA C.3.

$$\begin{aligned} & \sum_{0 \leq a \leq m, b=m-a} (a+1) \frac{\binom{c_<}{a} \binom{c_<+c_= - a}{b}}{\binom{a+b}{a}} \\ & = \frac{(m+1)(m+2)}{(c_= + 1)(c_= + 2)} \binom{c_<+c_= + 2}{m+2} - \frac{(c_< + 1)(c_< - m)}{c_= + 1} \binom{c_<}{m} \\ & \quad + \frac{(c_< - m)(c_< - m - 1)}{c_= + 2} \binom{c_<}{m} \end{aligned} \quad (28)$$

Using this lemma, the contribution of second term to the weight of $\nabla_\theta \pi_\theta(y|x)$ is

$$\begin{aligned} & -\frac{c_=}{(n-1)\binom{n-2}{k-2}} \left[\frac{(m+1)(m+2)}{(c_= + 1)(c_= + 2)} \binom{c_<+c_= + 2}{m+2} - \frac{(c_< + 1)(c_< - m)}{c_= + 1} \binom{c_<}{m} \right. \\ & \quad \left. + \frac{(c_< - m)(c_< - m - 1)}{c_= + 2} \binom{c_<}{m} \right] R(x, y) \end{aligned} \quad (29)$$

As the first term and the second term all involves $R(x, y)$, we sum up their contribution together and obtain:

$$\begin{aligned} & \frac{1}{\binom{n-1}{k-1}} \frac{k}{c_= + 1} [\binom{c_<+c_= + 1}{k} - \binom{c_<}{k}] R(x, y) \\ & - \frac{c_=}{(n-1)\binom{n-2}{k-2}} \left[\frac{(m+1)(m+2)}{(c_= + 1)(c_= + 2)} \binom{c_<+c_= + 2}{m+2} - \frac{(c_< + 1)(c_< - m)}{c_= + 1} \binom{c_<}{m} \right. \\ & \quad \left. + \frac{(c_< - m)(c_< - m - 1)}{c_= + 2} \binom{c_<}{m} \right] R(x, y) \\ & = k \frac{\binom{c_<}{k-1}}{\binom{n-1}{k-1}} R(x, y) \end{aligned} \quad (30)$$

Surprisingly, though the contribution of $R(x, y)$ is complex at the beginning, its final expression is very straightforward.

Putting together all three terms, we have

THEOREM C.4. When $n \geq k$, an unbiased estimator for Max@k is,

$$\begin{aligned} J_{max@k}^{RSPO}(\theta) &= \frac{k}{n|\mathcal{D}_b|} \sum_{x \sim \mathcal{D}_b} \sum_{i=1}^n [R(x, y^i) \frac{\binom{c_<[i]}{k-1}}{\binom{n-1}{k-1}} - \\ & \quad \frac{k-1}{n-1} \sum_{j=1}^{c_<[i]} R(x, y^j) \frac{\binom{j-1}{k-2}}{\binom{n-2}{k-2}}] \nabla_\theta \log \pi_\theta(y^i|x) \end{aligned} \quad (31)$$

where we sort $\{y^i\}_{i=1}^n$ by $R(x, y_1) \leq R(x, y_2) \leq \dots \leq R(x, y_n)$ and set $c_<[i] = |\{j | R(x, y^j) < R(x, y^i)\}|$.

After combinatorial operations, Theorem C.4 proposes a simple unbiased estimator for the Max@k object, which greatly simplifies the complex computation in Algorithm 2.

Moreover, compared with the Max@k estimator in Section 4.3 with approximations, the only difference is replacing $i-1$ by $c_<[i]$. This again verify that the estimator in Section 4.3 is almost correct in the continues reward scenarios.

In addition, when rewards are binary, only y^i with $R(x, y^i) = 1$ has non-zero gradient weight, and the $c_<[i]$ is the number of responses with $R(x, y) = 0$. This indicates that this Max@k estimator is exactly the Pass@k estimator of Theorem 4.3.