GVPO: Group Variance Policy Optimization for Large Language Model Post-Training

Kaichen Zhang[1,2] **Yuzhong Hong**[2] **Junwei Bao**[2,*] **Hongfei Jiang**[2]
Yang Song[2] **Dingqian Hong**[2] **Hui Xiong**[1,3,*]
[1]Thrust of Artificial Intelligence, Hong Kong University of Science and Technology (Guangzhou)
[2]Zuoyebang Education Technology [3]Department of Computer Science and Engineering, HKUST
kzhangbi@connect.ust.hk eugene.h.git@gmail.com baojunwei001@gmail.com
{jianghongfei,songyang,hongdingqian}@zuoyebang.com xionghui@ust.hk

Abstract

Post-training plays a crucial role in refining and aligning large language models to meet specific tasks and human preferences. While recent advancements in post-training techniques, such as Group Relative Policy Optimization (GRPO), leverage increased sampling with relative reward scoring to achieve superior performance, these methods often suffer from training instability that limits their practical adoption. As a next step, we present **Group Variance Policy Optimization (GVPO)**. GVPO incorporates the analytical solution to KL-constrained reward maximization directly into its gradient weights, ensuring alignment with the optimal policy. The method provides intuitive physical interpretations: its gradient mirrors the mean squared error between the central distance of implicit rewards and that of actual rewards. GVPO offers two key advantages: (1) it guarantees a **unique optimal solution**, exactly the KL-constrained reward maximization objective, (2) it supports flexible sampling distributions that **avoids importance sampling and on-policy limitations**. By unifying theoretical guarantees with practical adaptability, GVPO establishes a new paradigm for reliable and versatile LLM post-training.

1 Introduction

Large language models (LLMs) [59, 31], trained on extensive datasets, exhibit impressive general-purpose capabilities, yet their practical utility and alignment with human values depend critically on post-training [46] refinement. While pre-training [60, 33] equips LLMs with broad linguistic patterns, post-training techniques—such as supervised fine-tuning (SFT) [32] and reinforcement learning [30] from human feedback (RLHF) [3] are indispensable for adapting these models to specialized applications and ensuring their outputs align with ethical, safety, and user-centric standards.

> "The biggest lesson that can be read from 70 years of AI research is that general methods that leverage computation are ultimately the most effective, and by a large margin." — Rich Sutton, 2024 Turing Award winner

This principle outlined in *The Bitter Lesson* [43]—which advocates for scalable, computation-driven approaches—is exemplified by recent advances in post-training, particularly Group Relative Policy Optimization (GRPO) [39]. Diverging from conventional reinforcement learning frameworks [37] that depend on training a separate value function, GRPO directly optimizes advantage by standardizing reward scores across samples. This approach eliminates the need for an auxiliary value model, which typically demands computational resources comparable to those of the policy model itself. As a result, GRPO significantly reduces memory and computational overhead, enabling more efficient sampling and scalable training. Deepseek-R1 [11] leverages GRPO and achieves significant performance.

*Corresponding Authors. Code available at https://github.com/jszkc/GVPO

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

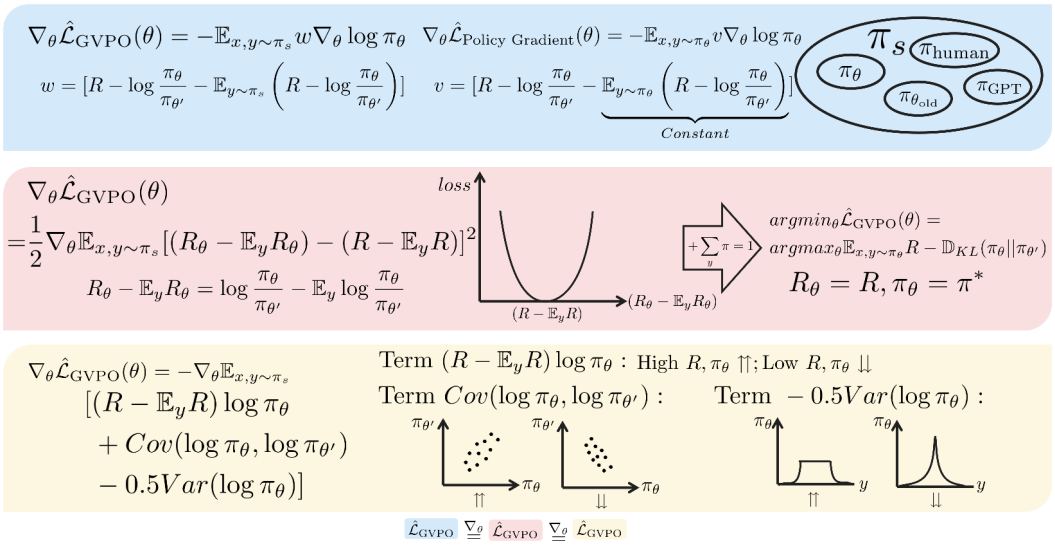

Figure 1: **Three equivalent loss functions of GVPO offer distinct interpretations**: (1) The Negative log-Likelihood perspective (top) illustrates that GVPO accommodates broader sampling distributions compared to conventional policy gradient methods; (2) The Mean Squared Error interpretation (middle) reveals GVPO's unique optimal solution, which simultaneously maximizes reward under a KL constraint; and (3) The Reinforcement Learning viewpoint (bottom) highlights GVPO's implicit regularization terms that ensure stable policy optimization. We assume $\beta = 1$ for simplicity.

However, GRPO has been documented to experience issues with training instability in prior literature [54, 28]. Specifically, GRPO is highly sensitive to its hyperparameters, such as the clip threshold and the KL-divergence coefficient. These limitations undermine the robustness of GRPO and hinder its broader practical adoption. As a next step, we propose **Group Variance Policy Optimization (GVPO)**, a novel approach for reliable and versatile LLM post-training.

Our analysis begins with a key observation: post-training algorithms—including but not limited to SFT, Reject Sampling [48], and GRPO—share a unified mathematical structure in their loss gradients [39, 10]. Specifically, each method's gradient can be expressed as a weighted sum of the gradients of the log-likelihoods of responses. This unified framework reveals that we can directly design weights to encode preferences—positive weights amplify gradients for favored responses, while negative weights suppress disfavored ones, with magnitudes modulating the strength of preference.

Motivated by the success of Direct Preference Optimization (DPO) [34]—which utilizes a closed-form link between reward models and the optimal policy under KL-divergence constraints [23]—we explore how to leverage this analytical relationship. A central obstacle arises from the partition function in the closed-form formula, which requires intractable expectation calculations over all possible responses. To address this, we identify a critical condition: *when the sum of assigned response weights within a prompt group equals zero, the partition function becomes invariant across compared responses*, effectively canceling out in the policy update rule. This insight eliminates the need for explicit estimation of the partition function, thereby enabling deployment of the closed-form optimal policy while retaining its theoretical advantages.

Based on the previous findings, we design GVPO's weighting scheme where the gradient weight of a response in a group is the difference between the central distance of implicit rewards-which derive from the current policy and the reference policy-and that of actual rewards, illustrated in Figure 1 (top panel). The loss is computable because the sum of weights in a prompt group equals zero.

We demonstrate that GVPO loss function carries physically meaningful interpretations. Specifically, we establish that its gradient equals that of a mean squared error loss measuring the discrepancy between implicit and actual reward central distances, illustrated in Figure 1 (middle panel).

Furthermore, the loss function in GVPO can be decomposed into three distinct components, as visualized in Figure 1 (bottom panel): (1) a group-relative reward term, (2) the variance of the current policy, and (3) the covariance between the current policy and a reference policy. The first component directly promotes advantage maximization by prioritizing responses with higher expected returns.

The covariance term acts as a regularizer, mitigating excessive deviations from the reference policy to ensure stable policy updates. Meanwhile, the variance term encourages moderate entropy, thereby naturally balancing exploration and exploitation. We systematically analyze GVPO's structural similarities with conventional policy gradient reinforcement learning methods.

We demonstrate that GVPO offers two key advantages:

- GVPO has a unique optimal solution, which coincides precisely with the optimal solution of the KL-constrained reward maximization. This guarantee confers a significant theoretical advantage over DPO. Prior work [4, 16] highlights that DPO may fail to converge to the optimal policy for the KL-constrained reward maximization problem, because of the inherent flaw of Bradley-Terry model [52]. In contrast, GVPO guarantees that its loss function is aligned with the original constrained optimization problem, ensuring convergence to the globally optimal policy. This theoretical robustness positions GVPO as a more reliable method for policy optimization.
- GVPO supports flexible sampling distributions that avoids importance sampling and on-policy limitations. Beyond the common practice of sampling from the previous step's policy, GVPO retains theoretical guarantees for the unique optimal solution under any sampling distribution satisfying a mild condition. This property provides a notable theoretical advantage over policy gradient methods [44]. Unlike on-policy approaches [41, 50], which require fresh trajectories for updates, GVPO facilitates off-policy training using reusable or heterogeneous datasets. Furthermore, in contrast to off-policy methods [39, 37] reliant on importance sampling, GVPO inherently avoids gradient explosion risks without introducing bias through clipping techniques.

As a result, GVPO emerges as a competitive online RL algorithm, capable of leveraging diverse data sources, sustaining stable policy updates, and preserving convergence to optimality.

2 Preliminary

Large language models take a prompt x as input and generate a response y as output. A policy $\pi_\theta(y_t|x, y_{<t})$ with parameter θ maps a sequence of tokens generated (x and $y_{<t}$) to a probability distribution over the next token y_t. We also denote $\pi_\theta(y|x)$ as the probability of generating the response y from x. A reward model $R(x, y)$ scores the response y as the reply to the prompt x.

A reward model can explicitly be evaluation ratings of human beings; or a trainable function that implicitly reflects human preferences; or a predefined rule, such as correctness, accuracy.

The general purpose of post-training of large language model is summarized as following: Given an initial policy $\pi_{\theta_{init}}$, a dataset of prompts $x \sim D$, a reward model R, the objective is to train a new policy π_θ that generates responses with higher rewards, that is, maximize $E_{x\sim D, y\sim\pi_\theta(\cdot|x)} R(x, y)$.

2.1 Towards better computation leverage in post-training

The initial stage of large language model post-training typically involves Supervised Fine-Tuning (SFT) [32]. In this phase, a dataset comprising input prompts x paired with exemplary responses y is used to optimize the pre-trained model. The training minimizes the negative log-likelihood loss:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\sum_{(x,y)\in\mathcal{D}} \log \pi_\theta(y|x) \tag{1}$$

Recent advancements, such as GRPO [39, 56], better leverage the computation by incorporating multiple sampled responses with their standardized reward scores as weights:

$$\mathcal{L}_{GRPO}(\theta) = -\sum_{(x,\{y_i\})\in\mathcal{D}} \sum_{i=1}^{k} Clip(\frac{\pi_\theta(y_i|x)}{\pi_{\text{old}}(y_i|x)}) \frac{R_i - \overline{R}}{\sigma(R)} \log \pi_\theta(y_i|x) - \beta KL(\pi_\theta||\pi_{\text{ref}}) \tag{2}$$

Rewards below average lead to negative weights, minimizing their likelihoods. However, minimizing likelihood can result in unstable training [1]. Additionally, off-policy training of GRPO, which involves importance sampling with the weight $\frac{\pi_\theta}{\pi_{\theta_{old}}}$, becomes unstable when π_θ significantly deviates from $\pi_{\theta_{\text{ref}}}$, potentially causing gradient explosions. To mitigate these issues, GRPO uses gradient clipping and a KL constraint between the updated and reference policies. Nevertheless, empirical results [54, 17] show that GRPO still exhibits training instability, which undermines its performance.

2.2 Optimal solution to the KL-constrained reward maximization

In human preference alignment scenario [32], the ideal reward model would directly reflect human evaluative judgments. However, obtaining explicit human ratings is often unavailable in practice. Instead, contemporary approaches typically leverage pairwise response preferences (x, y_w, y_l), where y_w denotes the preferred response and y_l the dispreferred response to prompt x, to approximate human preferences through reward model training. The resulting reward model subsequently enables policy optimization through the following KL-regularized objective:

$$max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(y|x)}[R(x,y)] - \beta\mathbb{D}_{KL}[\pi_\theta(y|x)||\pi_{\theta'}(y|x)] \tag{3}$$

where $\beta > 0$ controls the divergence penalty from policy $\pi_{\theta'}$. In preference alignment scenario, $\pi_{\theta'}$ is set to a reference policy π_{ref}.

Rather than employing separate reward modeling and policy optimization stages, DPO [34] derives a single-stage training paradigm by exploiting the analytical relationship between optimal policies and reward functions. The optimal solution to Equation 3 satisfies:

$$\pi^*(y|x) = \frac{1}{Z(x)}\pi_{\theta'}(y|x)e^{R(x,y)/\beta} \tag{4}$$

which implies the corresponding reward function:

$$R(x,y) = \beta \log \frac{\pi^*(y|x)}{\pi_{\theta'}(y|x)} + \beta \log Z(x) \tag{5}$$

where $Z(x) = \sum_y \pi_{\theta'}(y|x)e^{R(x,y)/\beta}$ represents the partition function. DPO circumvents explicit computation of $Z(x)$ by substituting the reward expression from Equation 5 into the Bradley-Terry loss [5], yielding the final objective:

$$\mathcal{L}_{\text{DPO}}(\theta) = -\sum_{(x,y_w,y_l)\in\mathcal{D}} \log\sigma(\beta \log \frac{\pi_\theta(y_w|x)}{\pi_{ref}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{ref}(y_l|x)}) \tag{6}$$

The success of DPO has been proven to be both efficient and effective. We attribute its achievements to its direct incorporation of the optimal policy's closed-form solution into the training objective.

2.3 Unified framework of post-training

As far as we know, post-training algorithms share a unified framework [39, 10], in which their losses' gradients share a same format:

$$\nabla_\theta\mathcal{L}(\theta) = -\sum_{(x,y_1,y_2,..,y_k)\in\mathcal{D}} \sum_{i=1}^{k} w_i \nabla_\theta \log \pi_\theta(y_i|x) \tag{7}$$

SFT only has one response per prompt, and its $w_1 = 1$. GRPO's essential weights are the standard scores of its rewards in a prompt group. Though it is not obvious for DPO, its gradients also share the same format, in which $w_w = \sigma(\beta \log \frac{\pi_\theta(y_l|x)}{\pi_{ref}(y_l|x)} - \beta \log \frac{\pi_\theta(y_w|x)}{\pi_{ref}(y_w|x)})$ and $w_l = -w_w$. Such the unified framework of post-training holds, because of the chain rule of derivatives.

Algorithm 1 Group Variance Policy Optimization

Require: initial policy π_θ; prompt distribution $\mathcal{D}$; hyperparameter β
1: **for** step $= 1, \ldots, n$ **do**
2: Sample a batch $\mathcal{D}_b$ from $\mathcal{D}$
3: Update the old policy model $\pi_{\theta_{old}} \leftarrow \pi_\theta$
4: Sample k responses $\{y_i\}_{i=1}^k \sim \pi_s(\cdot|x)$ for each prompt $x \in \mathcal{D}_b$
5: Compute rewards $\{R(x, y_i)\}_{i=1}^k$ for every sampled response y_i and prompt x
6: Iteratively update policy π_θ by minimizing the GVPO loss (Equation 8, setting $\pi_{\theta'} = \pi_{\theta_{old}}$)
7: **end for**
8: **Return** π_θ

3 Group Variance Policy Optimization

3.1 Motivation

The unified post-training framework (Equation 7) indicates that response preferences can be directly incorporated through the assignment of weights w_i.

To determine appropriate weights, we draw inspiration from the success of DPO. In particular, we aim to exploit the closed-form relationship between rewards and the optimal solution to the KL-constrained reward maximization objective: $R_\theta(x, y) = \beta \log(\pi_\theta(y|x)/\pi_{\theta'}(y|x)) + \beta \log Z(x)$.

However, the closed-form formula contains a partition function $Z(x)$ that is expensive to estimate in practice, because the function requires calculating the expectation of all possible responses.

To address this issue, we identify a critical condition: when the sum of assigned response weights within a prompt group equals zero, $\sum_{i=1}^{k} w_i = 0$, the partition function becomes invariant across responses: $\sum_{i=1}^{k} w_i R_\theta(x, y_i) = \sum_{i=1}^{k} w_i \beta \log(\pi_\theta(y_i|x)/\pi_{\theta'}(y_i|x))$.

3.2 Method

Build on this insight, we propose Group Variance Policy Optimization (GVPO), whose gradient weight w_i is the difference between the central distance of implicit rewards-which derive from policy π_θ and policy $\pi_{\theta'}$-and that of actual rewards. Formally, GVPO's gradient $\nabla_\theta \mathcal{L}_{\text{GVPO}}(\theta) =$

$$-\beta \sum_{(x,\{y_i\})\in\mathcal{D}} \sum_{i=1}^{k} [(R(x, y_i) - \overline{R(x, \{y_i\})}) - \beta(\log \frac{\pi_\theta(y_i|x)}{\pi_{\theta'}(y_i|x)} - \overline{\log \frac{\pi_\theta(\{y_i\}|x)}{\pi_{\theta'}(\{y_i\}|x)}})] \nabla_\theta \log \pi_\theta(y_i|x) \tag{8}$$

where $\overline{R(x, \{y_i\})} = \frac{1}{k} \sum_{i=1}^{k} R(x, y_i)$, and $\overline{\log \frac{\pi_\theta(\{y_i\}|x)}{\pi_{\theta'}(\{y_i\}|x)}} = \frac{1}{k} \sum_{i=1}^{k} \log \frac{\pi_\theta(y_i|x)}{\pi_{\theta'}(y_i|x)}$. We note that GVPO's gradient satisfies $\sum_{i=1}^{k} w_i = 0$. Algorithm 1 shows our proposed algorithm.

We demonstrate that GVPO's objective carries physically meaningful interpretations:

$$\begin{aligned}
&\nabla_\theta \mathcal{L}_{\text{GVPO}}(\theta) \\
=& -\sum_{x,\{y_i\}} \sum_{i=1}^{k} [(R(x, y_i) - \overline{R(x, \{y_i\})}) - (R_\theta(x, y_i) - \overline{R_\theta(x, \{y_i\})})] \nabla_\theta \beta \log \pi_\theta(y_i|x) \\
=& -\sum_{x,\{y_i\}} \sum_{i=1}^{k} [(R(x, y_i) - \overline{R(x, \{y_i\})}) - (R_\theta(x, y_i) - \overline{R_\theta(x, \{y_i\})})] \nabla_\theta R_\theta(x, y_i) \\
=& -\sum_{x,\{y_i\}} \sum_{i=1}^{k} [(R(x, y_i) - \overline{R(x, \{y_i\})}) - (R_\theta(x, y_i) - \overline{R_\theta(x, \{y_i\})})] \nabla_\theta (R_\theta(x, y_i) - \overline{R_\theta(x, \{y_i\})}) \\
=& \frac{1}{2} \nabla_\theta \sum_{x,\{y_i\}} \sum_{i=1}^{k} [(R_\theta(x, y_i) - \overline{R_\theta(x, \{y_i\})}) - (R(x, y_i) - \overline{R(x, \{y_i\})})]^2
\end{aligned}$$

The first and second steps hold because $\beta \log Z(x)$ can cancel out. The second step holds because $\sum_{i=1}^{k} w_i \nabla_\theta \overline{R(x, \{y_i\})} = 0$. The third step holds because $\nabla_x f(x)^2 = 2f(x)\nabla_x f(x)$.

Essentially, we have established that GVPO's gradient mathematically equals that of a mean squared error loss measuring the discrepancy between implicit and actual reward central distances. Intuitively, when implicit rewards equal actual rewards or with a constant group shift, the GVPO's loss is minimized. This interpretation also implies that the response with higher actual rewards in a group is also encouraged to have higher implicit rewards, indicating higher $\log \frac{\pi_\theta(y_i|x)}{\pi_{\theta'}(y_i|x)}$.

Furthermore, by rearranging the mean squared error loss, we can derive a variance-based formulation, which represents the "**Variance**" term in the name GVPO:

$$\frac{1}{2} \nabla_\theta \sum_{x,\{y_i\}} \sum_{i=1}^{k} [(R_\theta(x, y_i) - R(x, y_i)) - \overline{R_\theta(x, \{y_i\}) - R(x, \{y_i\})}]^2$$

3.3 Theoretical guarantee

We show that GVPO has an unique optimal solution, and this unique optimal solution is exactly the optimal solution of reward maximization with KL constraint (Equation 4). Formally,

Theorem 3.1. *The unique optimal policy that minimizes $\hat{\mathcal{L}}_{GVPO}(\theta)$, defined as*

$$\hat{\mathcal{L}}_{GVPO}(\theta) = \mathbb{E}_{x\sim\mathcal{D}}\mathbb{E}_{y\sim\pi_s(\cdot|x)}[(R_\theta(x,y) - \mathbb{E}_{y\sim\pi_s}R_\theta(x,y)) - (R(x,y) - \mathbb{E}_{y\sim\pi_s}R(x,y))]^2 \quad (9)$$

, is given by $\pi_\theta(y|x) = \pi^*(y|x) = \frac{1}{Z(x)}\pi_{\theta'}(y|x)e^{R(x,y)/\beta}$ for $\pi_s = \pi_{\theta'}$.

We prove the theorem by establishing both necessity and sufficiency, provided in Appendix B.1.

Theorem 3.1 implies that the parameters minimizing $\hat{\mathcal{L}}_{\text{GVPO}}(\theta)$—guaranteed to be the sole global optimum—also maximize the expected rewards while maintaining proximity to a reference policy.

The uniqueness of the solution ensures the optimization landscape is well-behaved, avoiding suboptimal local minima and guaranteeing convergence to a single, interpretable policy that optimally balances reward maximization with behavioral consistency relative to the reference. Consequently, this theorem bridges GVPO's practical algorithmic performance with theoretical guarantees.

Theorem 3.2. *The Theorem 3.1 also holds for any sampling distribution π_s satisfying $\forall x, \{y|\pi_{\theta'}(y|x) > 0\} \subseteq \{y|\pi_s(y|x) > 0\}$.*

Beyond the conventional practice of sampling from the reference policy ($\pi_s = \pi_{\theta'}$), GVPO retains the theoretical guarantee of a unique optimal solution under any sampling distribution that satisfies a mild condition. This condition is readily met by any policy π where $\pi(y, x) > 0$, a criterion inherently fulfilled by contemporary LLM policies utilizing softmax decoding.

The Theorem 3.2 of GVPO opens a new methodological avenue for off-policy LLM post-training. Prior off-policy methods have relied heavily on importance sampling [47], which suffers from two key limitations: (1) when π_θ diverges substantially from π_s, the importance weight $\frac{\pi_\theta}{\pi_s}$ can become either excessively large or vanishingly small, destabilizing training; and (2) when sampling involves heuristic or non-parametric components, π_s becomes intractable to compute, thereby prohibiting techniques such as experience replay [9] in modern LLM post-training. In contrast, GVPO supports highly flexible off-policy sampling strategies while maintaining strong theoretical guarantees, enabling more robust and practical post-training paradigms for large language models.

3.4 Discussions with DPO

We begin by analyzing the foundational commonality between GVPO and DPO: both methods integrate the closed-form solution to the reward maximization problem under a KL divergence constraint into their training objectives. This integration establishes a direct relationship between the learned policy π_θ and the implicit reward function R_θ, yielding two key advantages:

- It ensures an optimization process that inherently respects the KL divergence constraint, thereby preventing excessive deviation of the policy π_θ from the reference policy π_{ref}.
- It reduces the joint optimization over policies and rewards to a simpler problem focused solely on rewards. The latter is more tractable, as it requires only aligning the implicit rewards $R_\theta(x, y)$ with the true reward function $R(x, y)$.

To design effective methods leveraging this closed-form solution, two critical insights emerge:

1. **Computational Tractability**: The method must avoid intractable terms such as the partition function $Z(x)$. For instance, a naive loss $\mathcal{L} = \sum(R_\theta(x, y) - R(x, y))^2$ fails because $R_\theta(x, y)$ implicitly depends on $Z(x)$, which is computationally infeasible to estimate. DPO circumvents this by adopting the Bradley-Terry preference model, where $Z(x)$ cancels out in pairwise comparisons. GVPO proposes a novel *zero-sum property* across groups of responses, enabling cancellation of $Z(x)$ in broader multi-sample scenarios.
2. **Alignment with Desired Optimality**: The loss function must enforce meaningful convergence. For example, minimizing $\mathcal{L} = \sum\left(\beta\log\frac{\pi_\theta(x,y)}{\pi_{ref}(x,y)} - R(x, y)\right)^2$ yields a suboptimal solution

$R_\theta(x, y) = R(x, y) + \beta \log Z(x)$, which deviates from the true reward $R(x, y)$. A well-designed objective must avoid such misalignment. The method should adapt to available supervision. DPO leverages pairwise preference data without explicit rewards, while GVPO generalizes to group-wise responses with reward signals.

Moreover, GVPO demonstrates stronger theoretical robustness compared to DPO:

- Prior work [4, 16] highlights that DPO may fail to converge to the optimal policy for the KL-constrained reward maximization problem, because of the inherent flaw of Bradley-Terry model [52]. This arises because the DPO loss admits multiple minimizers, and its correlation with the true reward objective can diminish during training [45].
- In contrast, as formalized in Theorem 3.1 and Theorem 3.2, GVPO guarantees that its loss function is aligned with the original constrained optimization problem, ensuring convergence to the globally optimal policy. This theoretical robustness positions GVPO as a more reliable method for policy optimization in practice.

3.5 Discussions with GRPO and Policy Gradient methods

Structural similarities. Seeing the forest for the tree, we compare GVPO not only with GRPO but also with the broader family of policy gradient-based RL methods, beginning with their superficial structural similarities.

For simplicity, we assume $\beta = 1$ without loss of generality. Then $\hat{\mathcal{L}}_{\text{GVPO}}(\theta) \overset{\nabla_\theta}{=}$

$$\begin{aligned}
&\mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_s(\cdot|x)}[(R_\theta(x, y) - \mathbb{E}_y R_\theta(x, y))^2 - 2(R(x, y) - \mathbb{E}_y R(x, y))R_\theta(x, y)] \\
\overset{\nabla_\theta}{=}&\mathbb{E}_{x,y}[Var(\log \pi_\theta) - 2Cov(\log \pi_\theta, \log \pi_{\theta'}) - 2(R(x, y) - \mathbb{E}_y R(x, y)) \log \pi_\theta(y|x)] \\
=&-2\mathbb{E}_{x,y}[(R(x, y) - \mathbb{E}_y R(x, y)) \log \pi_\theta(y|x) + Cov(\log \pi_\theta, \log \pi_{\theta'}) - 0.5Var(\log \pi_\theta)]
\end{aligned} \tag{10}$$

where $Var(\log \pi_\theta) = (\log \pi_\theta(y|x) - \mathbb{E}_y \log \pi_\theta(y|x))^2$ and $Cov(\log \pi_\theta, \log \pi_{\theta'}) = (\log \pi_\theta(y|x) - \mathbb{E}_y \log \pi_\theta(y|x))(\log \pi_{\theta'}(y|x) - \mathbb{E}_y \log \pi_{\theta'}(y|x))$. As shown in Equation 10,

- the term $(R(x, y) - \mathbb{E}_y R(x, y)) \log \pi_\theta(y|x)$ encourages advantage maximization. Unlike conventional policy gradient methods that rely on explicit value function approximation [36], GRPO directly optimizes advantage by standardizing reward scores across samples. A distinction lies in GVPO's omission of standard deviation normalization.[2] Prior research [29] has also demonstrated that such scaling introduces bias by conflating prompt-level difficulty with reward signals.
- the term $Cov(\log \pi_\theta, \log \pi_{\theta'})$ serves to constrain deviations of the policy π_θ from policy $\pi_{\theta'}$, corresponding to $\mathbb{D}_{\text{KL}}[\pi_\theta || \pi_{\theta'}]$. Moreover, in GVPO, where $\pi_{\theta'} = \pi_{\theta_{\text{old}}}$, this term essentially aligns with the trust-region constraint [35], that ensures robustness between policy updates.
- the term $Var(\log \pi_\theta)$ strikes a balance between exploration and exploitation. We juxtapose this term with entropy regularization $-\mathbb{E}_y \log \pi(y|x)$ [2].
  - Increasing entropy encourages diversity by driving the policy toward a uniform distribution, but risks suppressing the likelihood of high-quality responses. Conversely, reducing entropy concentrates probability mass on a narrow set of outputs, diminishing diversity and potentially inducing entropy collapse. Consequently, entropy regularization proves highly sensitive to its coefficient, complicating practical implementation.
  - In contrast, $Var(\log \pi_\theta)$ circumvents this issue without requiring ad-hoc tuning by enabling scenarios where some responses receive zero probability, while other responses retain comparable probabilities. As the term $R - \overline{R}$ maximizes advantage, undesirable responses will receive zero probability, while favorable responses will maintain similar probabilities.
  - To illustrate the benefit, consider a toy example involving generation over the tokens a, b, c, d, e, where $\pi_\theta(a) = \pi_\theta(b) = 0$ and $\pi_\theta(c) = \pi_\theta(d) = \pi_\theta(e) = 1/3$. In this case, $Var(\log(\pi_\theta)) = 0$, indicating minimum variance and thus no penalty on this distribution. In contrast, entropy regularization only reaches its minimum either when the distribution is uniform or when it is one-hot, depending on whether one encourages higher or lower entropy.

[2] This omission is not a heuristic design but rather a consequence of the theoretical formulation.

In-depth similarities. In addition to structural similarities, we now highlight their key theoretical and practical distinctions. Modern policy gradient methods [35, 37, 39] optimize the expected reward under the current policy π_θ while constraining updates to avoid excessive deviation from the previous policy $\pi_{\theta_{\text{old}}}$. This is typically achieved by optimizing a objective that combines the reward $R(x, y)$ and a KL-divergence penalty term $\mathbb{D}_{\text{KL}}[\pi_\theta||\pi_{\theta_{\text{old}}}]$, yielding the gradient expression:

$$\begin{aligned}\nabla_\theta[\mathbb{E}_{x,y\sim\pi_\theta(y|x)}[R(x,y)] - \mathbb{D}_{\text{KL}}[\pi_\theta||\pi_{\theta_{\text{old}}}]] &= \nabla_\theta \mathbb{E}_x \sum_y \pi_\theta(y|x)(R(x,y) - \log\frac{\pi_\theta(y|x)}{\pi_{\theta_{\text{old}}}(y|x)}) \\ &= \mathbb{E}_{x,y\sim\pi_\theta(y|x)}(R(x,y) - \log\frac{\pi_\theta(y|x)}{\pi_{\theta_{\text{old}}}(y|x)} - 1)\nabla_\theta \log \pi_\theta(y|x)\end{aligned} \tag{11}$$

However, estimating this expectation requires on-policy sampling from $\pi_\theta(y|x)$, leading to low sample efficiency—a well-documented limitation of policy gradient methods. Reusing stale samples from prior policies introduces bias, degrading optimization stability and final performance.

To mitigate this, prior works [35, 37, 39] employ importance sampling, rewriting Equation 11 as:

$$\mathbb{E}_{x,y\sim\pi_{\theta_{\text{old}}}(y|x)}\frac{\pi_\theta(y|x)}{\pi_{\theta_{\text{old}}}(y|x)}\left(R(x,y) - \log\frac{\pi_\theta(y|x)}{\pi_{\theta_{\text{old}}}(y|x)} - 1\right)\nabla_\theta \log \pi_\theta(y|x). \tag{12}$$

This allows off-policy gradient estimation using samples from $\pi_{\theta_{\text{old}}}$. However, $\frac{\pi_\theta(y|x)}{\pi_{\theta_{\text{old}}}(y|x)}$ becomes unstable when π_θ deviates significantly from $\pi_{\theta_{\text{old}}}$, risking gradient explosion. Heuristics like gradient clipping [37] address this at the cost of biased gradient estimates, undermining theoretical guarantees.

GVPO circumvents these issues because it does not necessitate on-policy sampling in the first place. By rearranging Equation 11, we observe that the policy gradient can be expressed as:

$$\mathbb{E}_{x,y\sim\pi_\theta(y|x)}\left[R(x,y) - \log\frac{\pi_\theta(y|x)}{\pi_{\theta_{\text{old}}}(y|x)} - \mathbb{E}_{y\sim\pi_\theta(y|x)}\left(R(x,y) - \log\frac{\pi_\theta(y|x)}{\pi_{\theta_{\text{old}}}(y|x)}\right)\right]\nabla_\theta \log \pi_\theta(y|x),$$

where the baseline term (subtracted expectation) arises because $\mathbb{E}_{y\sim\pi_\theta}[c\nabla_\theta \log \pi_\theta(y|x)] = \nabla_\theta c = 0$ for any constant c. Crucially, the GVPO gradient generalizes this structure, $\nabla_\theta \hat{\mathcal{L}}_{\text{GVPO}}(\theta) =$

$$\mathbb{E}_{x,y\sim\pi_s(y|x)}\left[R(x,y) - \log\frac{\pi_\theta(y|x)}{\pi_{\theta_{\text{old}}}(y|x)} - \mathbb{E}_{y\sim\pi_s(y|x)}\left(R(x,y) - \log\frac{\pi_\theta(y|x)}{\pi_{\theta_{\text{old}}}(y|x)}\right)\right]\nabla_\theta \log \pi_\theta(y|x),$$

This reveals that classical policy gradient under trust-region constraint is a special case of GVPO gradient with $\pi_s = \pi_\theta$. As proven in Theorem 3.1 and Theorem 3.2, GVPO retains the same optimal solution as the policy gradient method while decoupling the sampling distribution π_s from the learned policy π_θ. GVPO's decoupling addresses two critical limitations:

1. **Sample Efficiency**: Unlike on-policy methods [41, 50], GVPO supports off-policy training with reusable or mixed data (e.g., expert demonstrations, historical policies, or model distillations).
2. **Stability**: By avoiding importance sampling weights $\frac{\pi_\theta}{\pi_{\theta_{\text{old}}}}$, GVPO eliminates gradient explosion risks without biased clipping.

By synergizing these advantages, GVPO emerges as a competitive online reinforcement learning algorithm capable of leveraging diverse data sources, sustaining stable policy updates, and preserving convergence to optimality—a combination previously unattained in prior policy gradient methods.

Table 1: Algorithm Performance Comparison on Mathematical Datasets

Algorithm	AIME2024	AMC	MATH500	Minerva	Olympiadbench
Qwen2.5-Math-7B	14.68	38.55	64.00	27.20	30.66
+GRPO	14.79	55.42	80.00	41.17	42.07
+Dr.GRPO	16.56	48.19	81.20	44.48	43.40
+Remax	17.19	60.24	82.00	40.44	45.19
+Reinforce++	16.67	54.22	80.40	43.01	41.78
+GVPO	**20.72**	**62.65**	**83.80**	**45.95**	**46.96**

4 Experiments

Task. Following the established experimental setting of GRPO, we conduct a comprehensive evaluation on math reasoning. Specifically, we post-train the Qwen2.5-Math-7B model on Competition Math dataset [14] and assess performance on AIME2024 [26], AMC [26], Math500 [15], Minerva [25], and OlympiadBench [13]. For answer verification, we utilize the xVerify framework [6]. We adopt the $pass@1$ accuracy for all benchmarks except AIME2024, where we report $avg@32$ accuracy to account for its limited size (30 problems) and high difficulty.

In addition to math reasoning, we also evaluate GVPO on the summarization task in Appendix C.

Setup. To ensure a fair comparison across methods, we maintain identical experimental settings while only modifying the algorithmic component. For GVPO, we employ $\beta = 0.1$ and $\pi_s = \pi_{\theta_{\text{old}}}$ in the main experiment. For competing approaches, we utilize hyperparameters specified in their original publications. All experiments generate $k = 5$ responses per prompt. A comprehensive description of the training details is provided in Appendix A.1.

Main Result. Table 1 shows the main experiment result, which demonstrates that GVPO achieves the best performance, outperforming both the base model and other variants in all benchmarks [28, 27, 17], particularly in complex problem-solving scenarios. We attribute its effectiveness to its strong theoretical guarantees of convergence.

Ablation on β. Figure 2 analyzes the sensitivity of GVPO to variations in β. The results demonstrate little fluctuation in performance across β, suggesting GVPO exhibits robustness to this hyperparameter. This stability may reduce the need for exhaustive tuning and enhance its practical utility.

Ablation on k. Figure 3 examines how GVPO scales with k, evaluated on Qwen2.5-Math-1.5B. Top and bottom panels show results for MATH500 and AIME2024 respectively. GVPO consistently outperforms GRPO across all k and demonstrates superior scalability. Notably, GVPO matches the AIME2024 performance of a 7B model on the 1.5B architecture through increased k, highlighting its potential for reducing inference costs in practice.

Ablation on π_s. Figure 4 investigates GVPO's versatility on sampling distributions, evaluated on Qwen2.5-Math-1.5B and MATH500. We propose a heuristic π_s that mixes responses from $\pi_{\theta_{\text{old}}}$ with historical responses. Results demonstrate GVPO's robust performance across mixing proportions, highlighting: (1) this π_s can reduce sampling costs during training, and (2) it suggests GVPO's potential to bridge modern LLM research with previous RL research on exploration strategies.

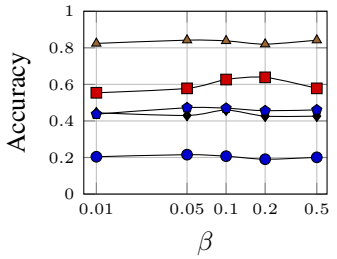

Figure 2: Ablation on β. Each line represents a dataset.

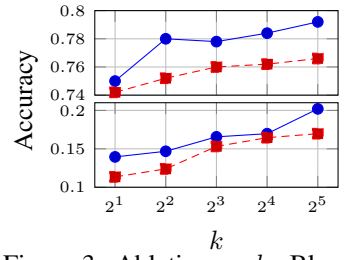

Figure 3: Ablation on k. Blue line: GVPO; Red line: GRPO.

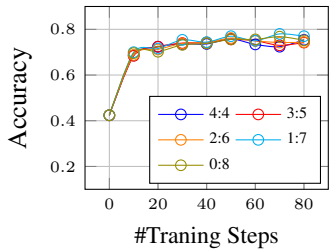

Figure 4: Ablation on π_s. #(historical y) : #(y from $\pi_{\theta_{\text{old}}}$)

Table 2: Ablation on Regularization Terms

Algorithm	AIME24	AMC	MATH500	Minerva	Olympiadbench
GVPO	20.72	62.65	83.80	45.95	46.96
GVPO - Var	0.00	0.00	0.00	0.00	0.00
GVPO - Cov	0.00	0.00	0.00	0.00	0.00
GVPO - Var - Cov	7.19	39.76	73.00	34.93	35.70
GVPO - Var + Entropy	0.00	0.00	0.00	0.00	0.00
GVPO - Var + Entropy (LR=1e-6)	3.02	39.76	33.20	23.16	8.89

Ablation on Regularization Terms. In Section 3.5, we decompose the GVPO loss into three components: an advantage maximization term, a variance regularization term (Var) on the current policy, and a covariance regularization term (Cov) between the current and reference policies.

Table 2 presents the ablation results for these regularization terms. We first remove $Var(\log \pi_\theta)$ and $Cov(\log \pi_\theta, \log \pi_{\theta'})$ individually. In both cases, the model fails to converge and generates incoherent outputs, indicating that each term plays a crucial role in stabilizing training. When both regularization terms are removed simultaneously—reducing the objective to $R - \overline{R}$—the model initially converges but diverges after approximately 10% of the training steps, further confirming that these regularization components are essential to GVPO's stability.

Study on $Var(\log \pi_\theta)$. We further investigate the role of $Var(\log \pi_\theta)$ by replacing it with an entropy regularization term. As shown in Table 2, the model again fails to converge and produces incoherent outputs. Lowering the learning rate to $1e{-}6$ improves stability marginally, yet substituting it with entropy regularization still lead to divergence or suboptimal performance. These findings suggest that although entropy regularization serves a similar stabilizing purpose, it cannot fully substitute $Var(\log \pi_\theta)$. The degradation likely arises from entropy regularization being either too weak—insufficient to suppress extreme updates—or too strong—impeding convergence. This underscores a key limitation of entropy regularization: its sensitivity to coefficient tuning. In contrast, the coefficient for $Var(\log \pi_\theta)$ in GVPO is derived analytically via the optimal solution theorem, eliminating the need for manual tuning and enhancing overall robustness.

Table 3: Algorithm Performance Comparison with Different Random Seeds

Algorithm	AIME2024	AMC	MATH500	Minerva	Olympiadbench
GRPO	13.59±1.11	44.34±2.19	76.30±1.35	35.40±1.04	38.65±0.74
GVPO	**15.56±1.05**	**45.78±2.05**	**77.36±0.98**	**36.03±1.15**	**39.58±1.08**

Robustness Check with Different Random Seeds. We conducted additional experiments using 10 random seeds for both methods on Qwen2.5-Math-1.5B. The results show that GVPO consistently outperforms GRPO in overall performance while exhibiting comparable standard deviations, indicating stable results across different runs.

Table 4: Algorithm Performance Comparison with Llama-3.1-8B-Instruct

Algorithm	AIME2024	AMC	MATH500	Minerva	Olympiadbench
Llama-3.1-8B-Instruct	5.00	19.27	50.40	22.42	17.04
+GRPO	8.54	19.27	52.60	25.37	16.15
+GVPO	**11.56**	**20.48**	**56.60**	**29.41**	**20.59**

Robustness Check with a Different Foundation Model. To further assess generalization, we repeated the experiments using Llama-3.1-8B-Instruct as the foundation model. As shown in Table 5, GVPO again outperforms GRPO, reaffirming the robustness and effectiveness of our approach.

5 Related Work

This paper closely relates to the LLM post-training literature, including SFT [32, 8], RLHF [3], PPO [37], TRPO [35], DPO [34], GRPO [39], Dr.GRPO [28], Remax [27], Reinforce++ [17], DAPO [54]; and debates on whether RL improves LLM reasoning capacity [55], spurious reward issue [38], data contamination issue [51], and Pass@K optimization [56].

Beyond post-training, LLMs have demonstrated broad applicability across domains, including vision-language modeling [12], graph learning [24, 19], AI safety research [53], evaluation frameworks leveraging LLM-as-a-Judge paradigms [22, 21], retrieval-augmented generation [7], AI for Education [20, 49] and economics of AI [57, 58]. These broader developments underscore the promising role of post-training in enhancing the adaptability of LLMs across a growing range of applications.

6 Conclusion

In this paper, we present Group Variance Policy Optimization (GVPO). GVPO guarantees a unique optimal solution, exactly the KL-constrained reward maximization objective. Moreover, it supports flexible sampling distributions that avoids on-policy and importance sampling limitations. Through systematic comparisons with other prominent methods both theoretically and empirically, we establish GVPO as a new paradigm for reliable and versatile LLM post-training.

Acknowledgments and Disclosure of Funding

We thank all the reviewers, the AC and program committee members for constructive feedback.

This work was supported in part by the National Key R&D Program of China (Grant No.2023YFF0725001), in part by the National Natural Science Foundation of China (Grant No.92370204), in part by the guangdong Basic and Applied Basic Research Foundation (Grant No.2023B1515120057), in part by the Education Bureau of Guangzhou.

References

[1] A. Abdolmaleki, B. Piot, B. Shahriari, J. T. Springenberg, T. Hertweck, R. Joshi, J. Oh, M. Bloesch, T. Lampe, N. Heess, et al. Learning from negative feedback, or positive feedback or both. arXiv preprint arXiv:2410.04166, 2024.

[2] Z. Ahmed, N. Le Roux, M. Norouzi, and D. Schuurmans. Understanding the impact of entropy on policy optimization. In International conference on machine learning, pages 151–160. PMLR, 2019.

[3] Y. Bai, A. Jones, K. Ndousse, A. Askell, A. Chen, N. DasSarma, D. Drain, S. Fort, D. Ganguli, T. Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. arXiv preprint arXiv:2204.05862, 2022.

[4] H. Bong and A. Rinaldo. Generalized results for the existence and consistency of the mle in the bradley-terry-luce model. In International Conference on Machine Learning, pages 2160–2177. PMLR, 2022.

[5] R. A. Bradley and M. E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. Biometrika, 39(3/4):324–345, 1952.

[6] D. Chen, Q. Yu, P. Wang, W. Zhang, B. Tang, F. Xiong, X. Li, M. Yang, and Z. Li. xverify: Efficient answer verifier for reasoning model evaluations, 2025.

[7] L. Dai, Y. Xu, J. Ye, H. Liu, and H. Xiong. Seper: Measure retrieval utility through the lens of semantic perplexity reduction. arXiv preprint arXiv:2503.01478, 2025.

[8] Y. Fan, Y. Hong, Q. Wang, J. Bao, H. Jiang, and Y. Song. Preference-oriented supervised fine-tuning: Favoring target model over aligned large language models. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 39, pages 23859–23867, 2025.

[9] W. Fedus, P. Ramachandran, R. Agarwal, Y. Bengio, H. Larochelle, M. Rowland, and W. Dabney. Revisiting fundamentals of experience replay. In International conference on machine learning, pages 3061–3071. PMLR, 2020.

[10] B. Gao, F. Song, Y. Miao, Z. Cai, Z. Yang, L. Chen, H. Hu, R. Xu, Q. Dong, C. Zheng, et al. Towards a unified view of preference learning for large language models: A survey. arXiv preprint arXiv:2409.02795, 2024.

[11] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv preprint arXiv:2501.12948, 2025.

[12] W. Guo, Z. Chen, S. Wang, J. He, Y. Xu, J. Ye, Y. Sun, and H. Xiong. Logic-in-frames: Dynamic keyframe search via visual semantic-logical verification for long video understanding. arXiv preprint arXiv:2503.13139, 2025.

[13] C. He, R. Luo, Y. Bai, S. Hu, Z. L. Thai, J. Shen, J. Hu, X. Han, Y. Huang, Y. Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. arXiv preprint arXiv:2402.14008, 2024.

[14] D. Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, D. Song, and J. Steinhardt. Measuring mathematical problem solving with the math dataset. arXiv preprint arXiv:2103.03874, 2021.

[15] D. Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, D. Song, and J. Steinhardt. Measuring mathematical problem solving with the math dataset. arXiv preprint arXiv:2103.03874, 2021.

[16] Y. Hong, H. Zhang, J. Bao, H. Jiang, and Y. Song. Energy-based preference model offers better offline alignment than the bradley-terry preference model. In Proceedings of the 42nd International Conference on Machine Learning, volume 267 of Proceedings of Machine Learning Research, pages 23787–23804. PMLR, 13–19 Jul 2025.

[17] J. Hu. Reinforce++: A simple and efficient approach for aligning large language models. arXiv preprint arXiv:2501.03262, 2025.

[18] J. Hu, Y. Zhang, Q. Han, D. Jiang, X. Zhang, and H.-Y. Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model, 2025.

[19] Z. Hu, Y. Li, Z. Chen, J. Wang, H. Liu, K. Lee, and K. Ding. Let's ask gnn: Empowering large language model for graph in-context learning. arXiv preprint arXiv:2410.07074, 2024.

[20] Z. Hu, J. Lian, Z. Xiao, S. Zhang, T. Wang, N. J. Yuan, X. Xie, and H. Xiong. Unveiling the learning mind of language models: A cognitive framework and empirical study. arXiv preprint arXiv:2506.13464, 2025.

[21] Z. Hu, L. Song, J. Zhang, Z. Xiao, T. Wang, Z. Chen, N. J. Yuan, J. Lian, K. Ding, and H. Xiong. Explaining length bias in llm-based preference evaluations. arXiv preprint arXiv:2407.01085, 2024.

[22] Z. Hu, J. Zhang, Z. Xiong, A. Ratner, H. Xiong, and R. Krishna. Language model preference evaluation with multiple weak evaluators. arXiv preprint arXiv:2410.12869, 2024.

[23] N. Jaques, S. Gu, D. Bahdanau, J. M. Hernández-Lobato, R. E. Turner, and D. Eck. Sequence tutor: Conservative fine-tuning of sequence generation models with kl-control. In International Conference on Machine Learning, pages 1645–1654. PMLR, 2017.

[24] W. Jiang, W. Wu, L. Zhang, Z. Yuan, J. Xiang, J. Zhou, and H. Xiong. Killing two birds with one stone: Cross-modal reinforced prompting for graph and language tasks. In Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pages 1301–1312, 2024.

[25] A. Lewkowycz, A. Andreassen, D. Dohan, E. Dyer, H. Michalewski, V. Ramasesh, A. Slone, C. Anil, I. Schlag, T. Gutman-Solo, et al. Solving quantitative reasoning problems with language models. Advances in Neural Information Processing Systems, 35:3843–3857, 2022.

[26] J. Li, E. Beeching, L. Tunstall, B. Lipkin, R. Soletskyi, S. Huang, K. Rasul, L. Yu, A. Q. Jiang, Z. Shen, et al. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. Hugging Face repository, 13:9, 2024.

[27] Z. Li, T. Xu, Y. Zhang, Z. Lin, Y. Yu, R. Sun, and Z.-Q. Luo. Remax: A simple, effective, and efficient reinforcement learning method for aligning large language models. arXiv preprint arXiv:2310.10505, 2023.

[28] Z. Liu, C. Chen, W. Li, P. Qi, T. Pang, C. Du, W. S. Lee, and M. Lin. Understanding r1-zero-like training: A critical perspective. arXiv preprint arXiv:2503.20783, 2025.

[29] Z. Liu, C. Chen, W. Li, P. Qi, T. Pang, C. Du, W. S. Lee, and M. Lin. Understanding r1-zero-like training: A critical perspective, 2025.

[30] R. Miao. Optimizing the Unknown: Reinforcement Learning and Energy-Based Model for Black Box Bayesian Optimization. University of California, Los Angeles, 2024.

[31] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, and J. Gao. Large language models: A survey, 2025.

[32] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al. Training language models to follow instructions with human feedback. Advances in neural information processing systems, 35:27730–27744, 2022.

[33] J. Qian, Z. Zhu, H. Zhou, Z. Feng, Z. Zhai, and K. Mao. Beyond the next token: Towards prompt-robust zero-shot classification via efficient multi-token prediction. arXiv preprint arXiv:2504.03159, 2025.

[34] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn. Direct preference optimization: Your language model is secretly a reward model. Advances in Neural Information Processing Systems, 36:53728–53741, 2023.

[35] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz. Trust region policy optimization. In F. Bach and D. Blei, editors, Proceedings of the 32nd International Conference on Machine Learning, volume 37 of Proceedings of Machine Learning Research, pages 1889–1897, Lille, France, 07–09 Jul 2015. PMLR.

[36] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel. High-dimensional continuous control using generalized advantage estimation. arXiv preprint arXiv:1506.02438, 2015.

[37] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347, 2017.

[38] R. Shao, S. S. Li, R. Xin, S. Geng, Y. Wang, S. Oh, S. S. Du, N. Lambert, S. Min, R. Krishna, et al. Spurious rewards: Rethinking training signals in rlvr. arXiv preprint arXiv:2506.10947, 2025.

[39] Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. Li, Y. Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300, 2024.

[40] G. Sheng, C. Zhang, Z. Ye, X. Wu, W. Zhang, R. Zhang, Y. Peng, H. Lin, and C. Wu. Hybridflow: A flexible and efficient rlhf framework. arXiv preprint arXiv: 2409.19256, 2024.

[41] S. Singh, T. Jaakkola, M. L. Littman, and C. Szepesvári. Convergence results for single-step on-policy reinforcement-learning algorithms. Machine learning, 38:287–308, 2000.

[42] N. Stiennon, L. Ouyang, J. Wu, D. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei, and P. F. Christiano. Learning to summarize with human feedback. Advances in neural information processing systems, 33:3008–3021, 2020.

[43] R. Sutton. The bitter lesson. Incomplete Ideas (blog), 13(1):38, 2019.

[44] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. Advances in neural information processing systems, 12, 1999.

[45] Y. Tang, Z. D. Guo, Z. Zheng, D. Calandriello, R. Munos, M. Rowland, P. H. Richemond, M. Valko, B. Á. Pires, and B. Piot. Generalized preference optimization: A unified approach to offline alignment. arXiv preprint arXiv:2402.05749, 2024.

[46] G. Tie, Z. Zhao, D. Song, F. Wei, R. Zhou, Y. Dai, W. Yin, Z. Yang, J. Yan, Y. Su, Z. Dai, Y. Xie, Y. Cao, L. Sun, P. Zhou, L. He, H. Chen, Y. Zhang, Q. Wen, T. Liu, N. Z. Gong, J. Tang, C. Xiong, H. Ji, P. S. Yu, and J. Gao. A survey on post-training of large language models, 2025.

[47] S. T. Tokdar and R. E. Kass. Importance sampling: a review. Wiley Interdisciplinary Reviews: Computational Statistics, 2(1):54–60, 2010.

[48] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288, 2023.

[49] T. Wang, Y. Zhan, J. Lian, Z. Hu, N. J. Yuan, Q. Zhang, X. Xie, and H. Xiong. Llm-powered multi-agent framework for goal-oriented learning in intelligent tutoring system. In Companion Proceedings of the ACM on Web Conference 2025, pages 510–519, 2025.

[50] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine learning, 8:229–256, 1992.

[51] M. Wu, Z. Zhang, Q. Dong, Z. Xi, J. Zhao, S. Jin, X. Fan, Y. Zhou, Y. Fu, Q. Liu, et al. Reasoning or memorization? unreliable results of reinforcement learning due to data contamination. arXiv preprint arXiv:2507.10532, 2025.

[52] W. Wu, B. W. Junker, and N. M. D. Niezink. Asymptotic comparison of identifying constraints for bradley-terry models, 2022.

[53] Y. Xu, A. Liu, X. Hu, L. Wen, and H. Xiong. Mark your llm: Detecting the misuse of open-source large language models via watermarking. arXiv preprint arXiv:2503.04636, 2025.

[54] Q. Yu, Z. Zhang, R. Zhu, Y. Yuan, X. Zuo, Y. Yue, T. Fan, G. Liu, L. Liu, X. Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. arXiv preprint arXiv:2503.14476, 2025.

[55] Y. Yue, Z. Chen, R. Lu, A. Zhao, Z. Wang, S. Song, and G. Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? arXiv preprint arXiv:2504.13837, 2025.

[56] K. Zhang, S. Gao, Y. Hong, H. Sun, J. Bao, H. Jiang, Y. Song, H. Dingqian, and H. Xiong. Rspo: Risk-seeking policy optimization for pass@ k and max@ k metrics in large language models. arXiv preprint arXiv:2508.01174, 2025.

[57] K. Zhang, Z. Yuan, and H. Xiong. The impact of generative artificial intelligence on market equilibrium: Evidence from a natural experiment. arXiv preprint arXiv:2311.07071, 2023.

[58] K. Zhang, Z. Yuan, and H. Xiong. Optimized cost per click in online advertising: A theoretical analysis. In Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pages 4232–4243, 2024.

[59] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J.-Y. Nie, and J.-R. Wen. A survey of large language models, 2025.

[60] C. Zhou, Q. Li, C. Li, J. Yu, Y. Liu, G. Wang, K. Zhang, C. Ji, Q. Yan, L. He, H. Peng, J. Li, J. Wu, Z. Liu, P. Xie, C. Xiong, J. Pei, P. S. Yu, and L. Sun. A comprehensive survey on pretrained foundation models: A history from bert to chatgpt, 2023.

Limitation

~~In compliance with page limit requirement, this paper does not extensively discuss experimental results. Instead, we have prioritized an in-depth analysis of GVPO's essence, as we posit that such a discussion offers greater scholarly value to readers. We acknowledge this structural limitation and note that it will be addressed in subsequent revisions, particularly in the event of acceptance with an additional page allowance.~~

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: In the abstract and introduction, we introduced GVPO and discussed GVPO in detail in the subsequent sections.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have a limitation section before the checklist.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We clarify the assumption along with each theoretical result, and provide detailed proofs in Appendix B.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We clearly introduced GVPO's loss and algorithm process in the main body of this paper, and showed how to implemented GVPO by using open-sources RL framework in Appendix A.2.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: In Appendix A.2, we show the code to implement GVPO and share instructions to reproduce experimental results in Appendix A.1.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Appendix A.1 provides a comprehensive description of the training details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

Justification: We do not have this result because training large language model is very costly.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We reported experiments compute resources in Appendix A.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We followed NeurIPS Code of Ethics in every respect.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: GVPO does not tie to particular applications, and does not lead to societal impacts.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: GVPO algorithm poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Yes, we credited them appropriately.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: We do not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development of GVPO does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

Appendix

A Supplementary Experiment Information

A.1 Experiment Settings

Hyperparameter Recipe. For each step, we sample 1024 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 and set the warm-up ratio to 5%. We grid-search the learning rate in $\{5e-7,\ 1e-6, 5e-6, 1e-5\}$ and find $5e-6$ to be the best setting. We conduct the main experiment using an Deepseek-R1-like chat template on top of Qwen2.5-Math-7B as in [18]. In the ablation experiments, for faster training and GPU memory limitations, we use the original Qwen chat template on top of Qwen2.5-Math-1.5B.

Compute Resources. We conduct our experiments using a server with eight 80GB H800 GPU cards. For Qwen2.5-Math-7B experiments with $k = 5$, it takes 6 to 8 minutes per training step and approximately 12 hours per experiment. For Qwen2.5-Math-1.5B experiments with $k = 8$, it takes 4 to 5 minutes per training step and approximately 8 hours per experiment.

A.2 Code Implementation

It is easy to implement GVPO based on open-source RL framework. For example[3], we show the minimum viable implementation of GVPO that only modifies a few line of GRPO loss in verl [40]:

```
def compute_policy_loss(old_log_prob, log_prob, advantages, eos_mask,
    **kwargs):
    scores = (log_prob * eos_mask).sum(dim=-1)
    scores_old = (old_log_prob * eos_mask).sum(dim=-1)
    advs = (advantages * eos_mask).sum(dim=-1) / eos_mask.sum(dim=-1)

    beta = 0.1
    k = scores.size(0)

    scores_new = scores.detach()
    loss = -1 * beta * scores * (advs - beta * ((scores_new -
        scores_new.mean()) - (scores_old - scores_old.mean())))

    return loss.sum()/(k-1)
```

Listing 1: A Simple GVPO Code Implementation

Moreover, we provide an implementation based on verl at https://github.com/jszkc/GVPO.

B Proofs

B.1 Proof of Theorem 3.1

Theorem 3.1. *The unique optimal policy that minimizes $\hat{\mathcal{L}}_{GVPO}(\theta)$, defined as*

$$\hat{\mathcal{L}}_{GVPO}(\theta) = \mathbb{E}_{x\sim\mathcal{D}}\mathbb{E}_{y\sim\pi_s(\cdot|x)}[(R_\theta(x,y) - \mathbb{E}_{y\sim\pi_s}R_\theta(x,y)) - (R(x,y) - \mathbb{E}_{y\sim\pi_s}R(x,y))]^2$$

, is given by $\pi_\theta(y|x) = \pi^*(y|x) = \frac{1}{Z(x)}\pi_{\theta'}(y|x)e^{R(x,y)/\beta}$ for $\pi_s = \pi_{\theta'}$.

Proof. We prove the theorem by establishing both necessity and sufficiency.

Necessity: If $\pi_\theta(y|x) = \pi^*(y|x)$, then it is an optimal policy solution.

The loss function $\hat{\mathcal{L}}_{\text{GVPO}}(\theta)$ is non-negative because it represents the expectation of a squared term:

$$\hat{\mathcal{L}}_{\text{GVPO}}(\theta) = \mathbb{E}_{x,y}\left[\left((R_\theta(x,y) - \mathbb{E}_y R_\theta(x,y)) - (R(x,y) - \mathbb{E}_y R(x,y)\right)^2\right] \geq 0.$$

[3]Make sure that 1) each input batch correspond to the k responses of a prompt and 2) the std normalizer in the GRPO advantage calculation has been removed.

When $\pi_\theta(y|x) = \pi^*(y|x)$, we have $R_\theta(x, y) = R(x, y)$. Substituting this into the loss function gives $\hat{\mathcal{L}}_{\text{GVPO}}(\theta) = 0$, confirming that π^* achieves the minimum loss.

Sufficiency: If a policy π_θ is optimal, then $\pi_\theta(y|x) = \pi^*(y|x)$.

Assume for contradiction that there exists an optimal policy $\pi_\theta \neq \pi^*$. Since π_θ is optimal, $\hat{\mathcal{L}}_{\text{GVPO}}(\theta) = 0$. This implies:

$$(R_\theta(x, y) - \mathbb{E}_y R_\theta(x, y)) = (R(x, y) - \mathbb{E}_y R(x, y)), \quad \forall x, y \text{ s.t. } \pi_s(y|x) > 0$$

Rewriting R_θ and R in terms of their respective policies:

$$\beta log \pi_\theta(y|x) - \mathbb{E}_y R_\theta(x, y) = \beta log \pi^*(y|x) - \mathbb{E}_y R(x, y).$$

Rearranging terms yields:

$$\pi_\theta(y|x) = \exp\left(\frac{\mathbb{E}_y[R_\theta(x, y) - R(x, y)]}{\beta}\right) \pi^*(y|x).$$

Since $\sum_{y \in \{y|\pi_{\theta'}(y|x)>0\}} \pi_\theta(y|x) = \sum_{y \in \{y|\pi_{\theta'}(y|x)>0\}} \pi^*(y|x) = 1$, we must have:

$$\sum_y \pi_\theta(y|x) = \exp\left(\frac{\mathbb{E}_y[R_\theta(x, y) - R(x, y)]}{\beta}\right) \sum_y \pi^*(y|x)$$

$$\implies \exp\left(\frac{\mathbb{E}_y[R_\theta(x, y) - R(x, y)]}{\beta}\right) = 1$$

Thus, $\pi_\theta(y|x) = \pi^*(y|x)$ for all x, y, contradicting the assumption $\pi_\theta \neq \pi^*$.

Since both necessity and sufficiency hold, the optimal policy is uniquely π^*. □

C Alternative Task on Summarization

To evaluate GVPO in the context of alignment [4], we conduct experiments on a summarization task following the experimental setup of DPO [34]. Specifically, we used the Reddit dataset from [42] and the reward model OpenAssistant/reward-model-deberta-v3-large-v2. First, we fine-tuned Qwen2.5 1.5B using the dataset to obtain an SFT reference model. We then applied both DPO and GVPO for post-training. The training data was sampled from the reference model and scored using the reward model.

For evaluation, we considered:

- The average reward of generated summaries on the test split.
- Win rate against human-written demonstration answers, as scored by the reward model.
- Preference accuracy using human-labeled comparison data: both DPO and GVPO are trained via $R_\theta(x, y) = \beta \frac{\log \pi_\theta(y|x)}{\log \pi_{\text{ref}}(y|x)} + \beta \log Z(x)$. Given a preference pair (y_w, y_l) where y_w is preferred, we compute whether $\frac{\log \pi_\theta(y_w|x)}{\log \pi_{\text{ref}}(y_w|x)} > \frac{\log \pi_\theta(y_l|x)}{\log \pi_{\text{ref}}(y_l|x)}$.
- Benchmarking by a set of powerful LLMs.
- Human evaluations by crowd-sourcing workers. We hire three workers to label their preference on 100 prompts on Prolific platform.

Table 5: Algorithm Performance Comparison on Summarization

Algorithm	Reward	Win Rate	Accuracy	Gpt-4o	Gemini-2.5-pro	Deepseek-R1	Human
SFT	2.84	41.85%	—	—	—	—	—
+DPO	4.83	68.28%	60.43%	31%	40%	25%	34%
+GVPO	5.75	79.49%	64.93%	69%	60%	75%	66%

The results indicate that GVPO outperforms DPO across these metrics, supporting the view that improvements in policy search methods are positively correlated with alignment quality.

[4] Thank Reviewer X8wt for sponsoring this study.