Listwise Policy Optimization: Group-based RLVR as Target-Projection on the LLM Response Simplex

Yun Qu[1,2]*, Qi Wang[1]†, Yixiu Mao[1], Heming Zou[1,2]*, Yuhang Jiang[1], Yingyue Li[1], Wutong Xu[1], Lizhou Cai[1], Weijie Liu[2], Clive Bai[2], Kai Yang[2], Yangkun Chen[2], Saiyong Yang[2]†, Xiangyang Ji[1]†
[1]Department of Automation, Tsinghua University
[2]LLM Department, Tencent

Abstract

Reinforcement learning with verifiable rewards (RLVR) has become a standard approach for large language models (LLMs) post-training to incentivize reasoning capacity. Among existing recipes, group-based policy gradient is prevalent, which samples a group of responses per prompt and updates the policy via group-relative advantage signals. This work reveals that these optimization strategies share a common geometric structure: each implicitly defines a target distribution on the response simplex and projects toward it via first-order approximation. Building on this insight, we propose Listwise Policy Optimization (LPO) to explicitly conduct the target-projection, which *demystifies the implicit target* by restricting the proximal RL objective to the response simplex, and then *projects the policy* via exact divergence minimization. This framework provides (i) monotonic improvement on the listwise objective with bounded, zero-sum, and self-correcting projection gradients, and (ii) flexibility in divergence selection with distinct structural properties through the decoupled projection step. On diverse reasoning tasks and LLM backbones, LPO consistently improves training performance over typical policy gradient baselines under matched targets, while intrinsically preserving optimization stability and response diversity. The code is available at https://github.com/thu-rllab/LPO.

1 Introduction

Recent advances have revealed the prominent potential of reinforcement learning with verifiable rewards (RLVR) for large language models (LLMs) post-training, which incentivizes reasoning capabilities on complex problem-solving tasks [Guo et al., 2025, Jaech et al., 2024, Luo et al., 2025]. In particular, critic-free, group-based RL paradigms, such as group relative policy optimization (GRPO) [Shao et al., 2024b], have been widely adopted for RLVR. This setup samples a group of responses, scores them with a verifier, and performs policy gradient updates using group-relative advantages. Further extensions in the literature [Liu et al., 2025b, Yu et al., 2025, Tajwar et al., 2026, Hu, 2025, Chen et al., 2025] have introduced critical refinements with special focus on advantage normalization and training stabilization.



Figure 1: LPO iteratively ascends the reward landscape via explicit target-projection, enabling stable optimization and flexible divergence design.

*Work completed during an internship at Tencent.
†Corresponding Authors: Qi Wang cheemswang@mail.tsinghua.edu.cn, Saiyong Yang stevesyang@tencent.com, Xiangyang Ji xyji@tsinghua.edu.cn

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Group-based policy gradients as implicit target-projections. Though these empirical refinements have proven effective, viewing them purely through the manner of advantage normalization obscures the intrinsic optimization mechanism. By defining a listwise distribution [Cao et al., 2007, Liu et al., 2025a] jointly over the sampled responses on a simplex, this work provides a unified geometric perspective on group-based RL algorithms: their advantage formulas implicitly construct a reward-weighted softmax target distribution over the responses, with the target's sharpness configured by the normalization scheme. Then, the standard policy gradient update acts merely as a first-order approximation of a reverse Kullback-Leibler (KL) [Kullback, 1951] projection toward this implicit target. This integrated perspective not only elucidates the workings of current methods but also motivates the explicit design of the target-projection mechanism.

From implicit approximation to explicit projection. Explicit target projection has been studied in classical RL [Peters et al., 2010, Abdolmaleki et al., 2018, Peng et al., 2019]. However, the existence of continuous action spaces necessitates the use of function approximation. In contrast, group-based RLVR exhibits a distinct and desirable property: the sampled responses for a prompt naturally form a finite simplex, allowing for the exact computation of both the target distribution and the projection in closed form. This makes it feasible to define clear separated goals between *what distribution to target* and *how to project toward it*, facilitating a seamless transition from implicit approximations to exact listwise optimization. Consequently, the central research question arises:

What properties emerge when this target-projection is made explicit, and how does this decoupled optimization space influence RLVR of LLMs?

Listwise Policy Optimization. In response to the above research question, this work develops Listwise Policy Optimization (LPO) to enable explicit target-projection on the response simplex. Specifically, LPO (i) explicates the implicit target by constraining the proximal RL objective to the sampled responses, yielding a closed-form solution with a controllable temperature, and (ii) optimizes the policy by projecting it onto the target via divergence minimization on the response simplex. The exact projection onto the simplex results in gradients that are bounded, zero-sum, and self-correcting by design, which induces variance reduction and stable optimization. Furthermore, the decoupled structure allows for flexible projection divergences, and we implement forward and reverse KL divergence as two representative instantiations. The resulting iterative target-projection algorithm provides provable monotonic improvement of the listwise reward per iteration.

Contributions. This work aims to offer deeper insights into policy optimization in RLVR, focusing on understanding and identifying potential improvements. The main contribution is two-fold:

1. We provide a unifying analytical perspective, revealing that group-based policy gradient methods implicitly perform approximate target-projections on the response simplex.
2. We develop LPO, an explicit target-projection framework that decouples listwise target construction from divergence projection, supported by theoretical analysis that proves improvement guarantee and characterizes projections' structural properties.

Extensive evaluations across logic, mathematics, programming, and multi-modal reasoning tasks with diverse LLM backbones demonstrate the effectiveness of LPO: (i) LPO achieves higher expected Pass@1 and Pass@k accuracy during training compared to baselines under matched implicit target constructions; (ii) decoupling the target from the projection accommodates diverse divergences, with a novel forward KL variant showing exceptional competitiveness; and (iii) LPO induces highly stable optimization trajectories while inherently preserving response diversity.

2 Preliminaries

2.1 Reinforcement Learning with Verifiable Rewards

RLVR has emerged as a critical post-training paradigm for incentivizing reasoning capabilities of LLMs [Shao et al., 2024b, Jaech et al., 2024]. Let x denote a prompt and $y = (y_1, \ldots, y_{|y|})$ a response of length $|y|$, generated autoregressively by a parameterized policy $\pi_\theta(y|x) = \prod_{i=1}^{|y|} \pi_\theta(y_i|x, y_{<i})$. Given a reward function $R(x, y)$ and a reference policy π_{ref}, the standard KL-regularized objective for RLVR [Shao et al., 2024b] is defined as:

$$J_x(\pi_\theta) = \mathbb{E}_{y\sim\pi_\theta(\cdot|x)}[R(x, y)] - \beta\, D_{\text{KL}}\big(\pi_\theta(\cdot|x) \,\|\, \pi_{\text{ref}}(\cdot|x)\big), \tag{1}$$

where $\beta \geq 0$ controls the strength of the reference constraint. Following recent advances [Yu et al., 2025, Qu et al., 2025b], we primarily focus on rule-based outcome rewards, which are typically binary or sparse ($R \in [0, 1]$), without an explicit reference penalty, i.e., $\beta = 0$.

2.2 Group-based Policy Gradient

The dominant paradigm in RLVR is group-based policy gradient (PG), represented by Group-Relative Policy Optimization (GRPO) [Shao et al., 2024b]. For each prompt x, a behavior policy π_b, which is typically the pre-update snapshot $\pi_{\theta_{\text{old}}}$, generates a group of K responses $\{y_1, \ldots, y_K\}$, each assigned a reward R_k forming the reward vector $R = [R_1, \ldots, R_K]^\top$. These rewards are converted into group-relative advantages, forming the advantage vector $A = [A_1, \ldots, A_K]^\top$ via centering and scaling. For instance, GRPO uses $A_k = \frac{R_k - \mu_G}{\sigma_G}$, where μ_G and σ_G are the group mean and standard deviation. Table 1 details other common normalization schemes. The policy is typically updated by maximizing a clipped surrogate objective [Schulman et al., 2017b, Shao et al., 2024b]:

$$\mathcal{L}_{\text{GRPO}}(\theta) = \mathbb{E}_{x, \{y_k\}_{k=1}^K \sim \pi_b(\cdot|x)} \left[\frac{1}{K} \sum_{k=1}^{K} \frac{1}{|y_k|} \sum_{i=1}^{|y_k|} \min\big(r_{k,i}\, A_k,\ \text{clip}(r_{k,i},\, 1-\epsilon,\, 1+\epsilon)\, A_k\big) \right], \quad (2)$$

where $r_{k,i}(\theta) = \frac{\pi_\theta(y_{k,i}|x, y_{k,<i})}{\pi_b(y_{k,i}|x, y_{k,<i})}$ is the importance ratio and ϵ is the clipping hyperparameter.

At the exact on-policy point ($\pi_\theta = \pi_b$), the importance ratios are identically one ($r_{k,i} = 1$). Consequently, for a fixed prompt x, the surrogate objective gradient reduces to the standard sequence-level group-based policy gradient [Sutton et al., 1999]:

$$g_{\text{PG}} = \frac{1}{K} \sum_{k=1}^{K} A_k \nabla_\theta \log \pi_\theta(y_k|x), \quad \text{where } \log \pi_\theta(y_k|x) \triangleq \frac{1}{|y_k|} \sum_{i=1}^{|y_k|} \log \pi_\theta(y_{k,i}|x, y_{k,<i}). \quad (3)$$

3 Group-based Policy Gradient as Implicit Target-Projection

This section reinterprets group-based policy gradients through the lens of the listwise distribution. We aim to explore: (i) the target distribution that these updates implicitly pursue, and (ii) the impact of different advantage normalization schemes on shaping that target.

3.1 Listwise Distribution on the Response Simplex

To formalize, we represent the policy's relative preference over the K sampled responses for prompt x as a listwise distribution P_θ [Cao et al., 2007, Rafailov et al., 2024, Liu et al., 2025a]:

$$P_{\theta,k} = \frac{\exp(s_{\theta,k})}{\sum_{j=1}^{K} \exp(s_{\theta,j})} = \text{softmax}(s_\theta)_k, \quad \text{with } s_{\theta,k} = \log \frac{\pi_\theta(y_k|x)}{\pi_b(y_k|x)}, \quad (4)$$

where P_θ reflects the extent to which π_θ prioritizes each response relative to π_b. At the on-policy point ($\pi_\theta = \pi_b$), P_θ reduces to the uniform distribution $1/K$. Since $P_{\theta,k} \geq 0$ and $\sum_k P_{\theta,k} = 1$, the vector P_θ lies on the probability simplex $\Delta^{K-1} = \{p \in \mathbb{R}^K : p_k \geq 0,\ \sum_k p_k = 1\}$, which we call the ***response simplex***.

3.2 Group-based Policy Gradient as Approximate Reverse KL

With the listwise distribution, we now reveal the underlying geometric property: standard group-based policy gradients implicitly perform target-projection via reverse Kullback-Leibler (KL) [Kullback, 1951] minimization.

Proposition 1 (Group-based policy gradient as reverse KL at on-policy)**.** *Let $A \in \mathbb{R}^K$ be a zero-mean advantage vector, i.e., $\sum_{k=1}^K A_k = 0$, and let $w^* = \text{softmax}(A)$. At the on-policy point ($\pi_\theta = \pi_b$), the policy gradient in Eq. (3) equals the negative gradient of the reverse KL divergence D_{KL}:*

$$g_{\text{PG}} = \frac{1}{K} \sum_{k=1}^{K} A_k \nabla_\theta \log \pi_\theta(y_k|x) = -\nabla_\theta D_{\text{KL}}(P_\theta \| w^*)\Big|_{\pi_\theta = \pi_b}. \quad (5)$$

Table 1: Advantages and implicit targets of existing group-based policy gradient methods.

Algorithm	Advantage A_k	Implicit target w^*	Temperature τ
Dr.GRPO [Liu et al., 2025b] / RLOO [Ahmadian et al., 2024]	$R_k - \mu_G$	$\mathrm{softmax}(R)$	~ 1
GRPO [Shao et al., 2024b] / DAPO [Yu et al., 2025]	$(R_k - \mu_G)/\sigma_G$	$\mathrm{softmax}(R/\sigma_G)$	σ_G
MaxRL [Tajwar et al., 2026]	$(R_k - \mu_G)/\mu_G$	$\mathrm{softmax}(R/\mu_G)$	μ_G

This observation identifies $w^* = \mathrm{softmax}(A)$ as the *implicit target* on the response simplex induced by the advantage design. This equivalence is exact at the on-policy point, but the approximation error grows as the policy drifts from the sampling distribution. Concretely, the per-response coefficient discrepancy scales as $\mathcal{O}(\bar{\delta} \cdot (1 + \|A\|_\infty)/K)$, where $\bar{\delta} = \max_k |\frac{\pi_\theta(y_k|x)}{\pi_b(y_k|x)} - 1|$ measures the degree of off-policy drift. See Appendix B.2 for detailed proof.

3.3 Implicit Targets of Existing Methods

Table 1 summarizes the specific implicit targets induced by existing group-based PG algorithms. Advantages in these methods take the form $A_k = (R_k - \mu)/\tau$ for various choices of centering μ and scaling τ. By the shift-invariance of softmax, the centering cancels and the target w^* reduces to $\mathrm{softmax}(R/\tau)$, where τ acts as a temperature. Different normalization schemes thus preserve the same reward ordering with the main difference in target sharpness, as detailed in Appendix C.3.

From approximation to exact projection. This unifying view also suggests a natural refinement. Since both the target w^* and the listwise distribution P_θ lie on the finite response simplex, the projection can be performed in an exact manner. Moreover, it provides a new lens on algorithm design worth investigating: exact projection allows for any statistical divergence, e.g., Forward KL, that were inaccessible under the current policy gradient paradigm. Accordingly, the next section will develop a generalized framework.

4 Listwise Policy Optimization

We now replace implicit policy gradient approximations with an explicit target-projection framework on the response simplex. This framework decouples each iteration into two entangled steps:

$$\underbrace{w^* = \arg\max_{w \in \Delta^{K-1}} \hat{J}(w)}_{\text{(i) Target: } \textit{what} \text{ distribution to aim for}} \qquad \underbrace{\theta' = \arg\min_\theta D(w^* \,\|\, P_\theta)}_{\text{(ii) Projection: } \textit{how} \text{ to optimize toward it}} \tag{6}$$

where $\hat{J}$ is a proximal objective on the simplex and D is a divergence measure. Next, we will detail the optimization steps, their implementation, and the theoretical analysis.

4.1 Target Induced on the Response Simplex

To demystify the principled origin of the implicit target in group-based policy gradients, we define a local proximal RL objective per prompt on the response simplex, which maximizes the expected reward subject to a trust region around the policy [Schulman et al., 2017a]:

$$\max_{w \in \Delta^{K-1}} \hat{J}(w) = \sum_{k=1}^{K} w_k R_k - \tau\, D_{\mathrm{KL}}(w \| P_t), \tag{7}$$

where $P_t = \mathrm{softmax}(s_t)$ is the listwise distribution induced by the pre-update policy π_t, with $s_{t,k} = \log\big(\pi_t(y_k|x)/\pi_b(y_k|x)\big)$. Equivalently, P_t is P_θ from Eq. (4) evaluated at $\theta = \theta_t$. Both P_t and s_t are held fixed while θ is updated.

Theorem 1 (Listwise Gibbs target)**.** *The objective $\hat{J}(w)$ in Eq. (7) has a unique maximizer w^*:*

$$w^*_k = \mathrm{softmax}(\phi)_k, \quad \textit{with } \phi_k = \frac{R_k}{\tau} + s_{t,k}. \tag{8}$$

Theorem 1 indicates that the target w^* re-weights the baseline P_t toward high-reward responses, with τ controlling the sharpness: $w^* \to \arg\max_k R_k$ as $\tau \to 0$, and $w^* \to P_t$ as $\tau \to \infty$. Under the on-policy setup ($\pi_t = \pi_b$), P_t degenerates to a uniform distribution and $w^* = \mathrm{softmax}(R/\tau)$ *recovers*

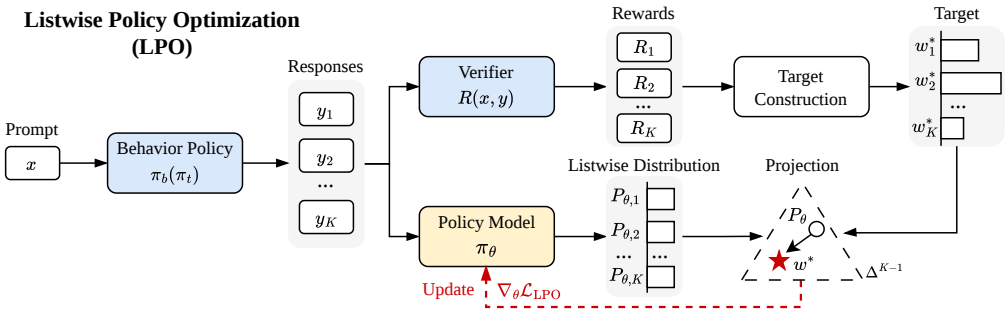


Figure 2: Illustration of LPO, which performs explicit target projection on the LLM response simplex, in contrast to the implicit approximations of group-based policy gradient methods.

the implicit targets of existing methods (Proposition 1), with τ now an explicit design parameter with trust-region interpretation rather than a byproduct of advantage normalization. Notably, concurrent works [Shu et al., 2026, Kaddour, 2026a] explore a similar paradigm by explicitly formulating the target as a reward-based Gibbs distribution with varied reference policies.

As $K \to \infty$, the empirical response simplex approximates the full policy space, and Eq. (7) recovers the KL-regularized RL objective $\max_w \mathbb{E}_w[R] - \tau D_{\text{KL}}(w\|\pi_t)$ [Ziebart, 2010, Levine, 2018], whose solution is $w^* \propto \pi_t(y)\exp(R(y)/\tau)$ with an intractable partition function. Operating on a finite response simplex yields a tractable formulation and makes the implicit target explicit.

Monotonic improvement guarantee. The proximal objective $\hat{J}(w)$ serves as a surrogate to the listwise reward $\hat{R}(w) = \sum_k w_k R_k$, establishing a performance improvement bound:

Theorem 2 (Performance improvement bound)**.** *Assume* $|R_k| \le R_{\max}$. *If the projection step achieves* $\text{TV}(P_{t+1}, w^*) \le \epsilon_{\text{proj}}$, *then*

$$\hat{R}(P_{t+1}) \ge \hat{R}(P_t) + \underbrace{\tau\big[D_{\text{KL}}(w^*\|P_t) + D_{\text{KL}}(P_t\|w^*)\big]}_{\textit{target gain} \ge 0} - \underbrace{2R_{\max}\epsilon_{\text{proj}}}_{\textit{projection error}}. \tag{9}$$

The target gain in Theorem 2 is the Jeffreys divergence [Jeffreys, 1946]. With perfect projection, i.e., $\epsilon_{\text{proj}} = 0$, the reward strictly improves whenever $P_t \neq w^*$. In the idealized full policy space, iterating the exact proximal update converges to the reward-maximizing policy, providing a limiting reference for the target-projection framework. See Appendix B.5 and Appendix B.6 for proofs.

Proposition 2 (Idealized full-space convergence)**.** *Let* $\pi_0(y) > 0$ *for all* y, *and assume* $R(y)$ *is bounded. Under exact proximal updates* $\pi_{t+1}(y) \propto \pi_t(y)\exp(R(y)/\tau)$, *the iteration satisfies* $\pi_t(y) \propto \pi_0(y)\exp(tR(y)/\tau)$ *and* $\mathbb{E}_{\pi_t}[R] \to \max_y R(y)$ *as* $t \to \infty$.

4.2 Projection for Policy Optimization

With both the target w^* in Eq. (8) and the listwise distribution P_θ in Eq. (4) on Δ^{K-1}, policy optimization reduces to a projection under a chosen divergence. As representative choices, we develop the forward and reverse KL versions, with full derivations in Appendix B.1.

Example 1 (**Forward KL**)**.** *Minimizing the forward KL divergence* $D_{\text{KL}}(w^*\|P_\theta)$ *gives:*

$$\min \mathcal{L}_{\text{LPO}_{\text{fwd}}} = D_{\text{KL}}(w^*\|P_\theta) \Rightarrow \nabla_\theta \mathcal{L}_{\text{LPO}_{\text{fwd}}} = \sum_{k=1}^{K} \underbrace{(P_{\theta,k} - w_k^*)}_{c_k^{\text{fwd}}} \nabla_\theta \log \pi_\theta(y_k|x). \tag{10}$$

The coefficient c_k^{fwd} measures the probability gap between the current policy and the target. Although similar projection objectives exist in prior methods [Abdolmaleki et al., 2018, Peng et al., 2019], they are implemented in a pointwise manner, treating each response independently without relative comparison. In contrast, LPO performs projection on the response simplex via shared normalization, which couples across responses. Furthermore, this yields the following desirable properties:

Corollary 1 (Gradient coefficient properties)**.** *The forward KL gradient coefficients* c_k^{fwd} *satisfy: (a) bounded:* $|c_k^{\text{fwd}}| \le 1$; *(b) zero-sum:* $\sum_k c_k^{\text{fwd}} = 0$; *(c) self-correcting:* $c_k^{\text{fwd}} \to 0$ *as* $P_\theta \to w^*$.

Algorithm 1 Listwise Policy Optimization (LPO)

Require: Policy parameters θ, temperature $\tau > 0$, batch size B, inner epochs E, step size η

for each training iteration **do**
  Set behavior policy $\pi_b \leftarrow \pi_\theta$, pre-update policy $\pi_t \leftarrow \pi_\theta$
  Sample a batch of prompts $\mathcal{B} = \{x_i\}_{i=1}^{B}$
  For each $x \in \mathcal{B}$, generate responses $\{y_k\}_{k=1}^{K} \sim \pi_b(\cdot|x)$ and compute rewards $\{R_k\}_{k=1}^{K}$
  Compute Target: $w^*(x) = \mathrm{softmax}(\phi(x))$ via Eq. (8) for all $x \in \mathcal{B}$
  for epoch $e = 1$ **to** E **do**
    Compute Coefficients: $c_k(x)$ via Eq. (10) (forward KL) or Eq. (11) (reverse KL)
    Gradient Update: $\theta \leftarrow \theta - \eta \frac{1}{B} \sum_{x \in \mathcal{B}} \sum_{k=1}^{K} c_k(x) \nabla_\theta \log \pi_\theta(y_k|x)$
  end for
end for

Corollary 2 (Mode-Coverage). *If $w_k^* \geq \alpha$ and $D_{\mathrm{KL}}(w^* \| P_\theta) \leq D$, then $P_{\theta,k} > \alpha \exp(-D/\alpha - 1)$.*

The zero-sum property acts as a built-in control variate for variance reduction [Sutton, 1988]. The bounded and self-correcting properties further improve optimization stability. Moreover, Corollary 2 provides a log-barrier against mode collapse, ensuring response diversity.

Example 2 (**Reverse KL**). *Minimizing the reverse KL divergence $D_{\mathrm{KL}}(P_\theta \| w^*)$, with logit gap $d_k = s_{\theta,k} - \phi_k$ (the difference between the current policy and the target) and its P_θ-weighted mean $\bar{d} = \sum_j P_{\theta,j}\, d_j$, yields the following gradient:*

$$\min \mathcal{L}_{\mathrm{LPO}_{\mathrm{rev}}} = D_{\mathrm{KL}}(P_\theta \| w^*) \Rightarrow \nabla_\theta \mathcal{L}_{\mathrm{LPO}_{\mathrm{rev}}} = \sum_{k=1}^{K} \underbrace{P_{\theta,k}\big(d_k - \bar{d}\big)}_{c_k^{\mathrm{rev}}} \nabla_\theta \log \pi_\theta(y_k|x). \tag{11}$$

Similar to the forward KL, the gradient coefficient c_k^{rev} is zero-sum and self-correcting. Minimizing reverse KL is equivalent to maximizing the proximal objective $\hat{J}$ (See Proposition 3), and it decomposes as $H(P_\theta) + \sum_k P_{\theta,k}\, \phi_k$: revealing an implicit entropy bonus (Appendix C.7). At the on-policy point, the gradient of this objective exactly recovers the standard policy gradient (Proposition 1).

4.3 Practical Implementation

The overall LPO procedure is summarized in Algorithm 1. The training pipeline is identical to standard group-based RL algorithms, with no additional computational cost.

Temperature as an adaptive baseline. While the temperature τ could theoretically be treated as a trust-region hyperparameter, we intentionally avoid introducing new tuning burdens. Instead, we adapt τ using the group-relative advantage normalization statistics of existing methods, e.g., $\tau = \sigma_G$ for GRPO or $\tau = \mu_G$ for MaxRL. This allows us to isolate gains from exact listwise projection while preserving the target temperature used by prior methods.

5 Main Empirical Results

5.1 Experimental Setup

We evaluate LPO across four representative domains of reasoning: **logic, mathematics, programming, and multi-modal geometry**. To assess generality, we benchmark across a **diverse set of LLM backbones** spanning different model sizes (1.5B–14B) and various LLM families. We track the training performance by plotting the curves for expected Pass@1 (average accuracy over rollouts) and Pass@k [Chen et al., 2021], with the specific k configurations detailed per benchmark.

Logical Reasoning. We adopt the Countdown Game, which requires composing given numbers using basic operations to match a target value. We train on a subset of Countdown-34 dataset [Pan et al., 2025] and evaluate on both Countdown-34 and the harder Countdown-4. We primarily use Qwen3-4B-Base [Yang et al., 2025a] and further evaluate models from other families in Sec. 5.4.3.

Mathematical Reasoning. We train on the MATH dataset [Hendrycks et al., 2021] using Qwen3-1.7B-Base and Qwen3-8B-Base [Yang et al., 2025a]. Evaluation is conducted on standard benchmarks

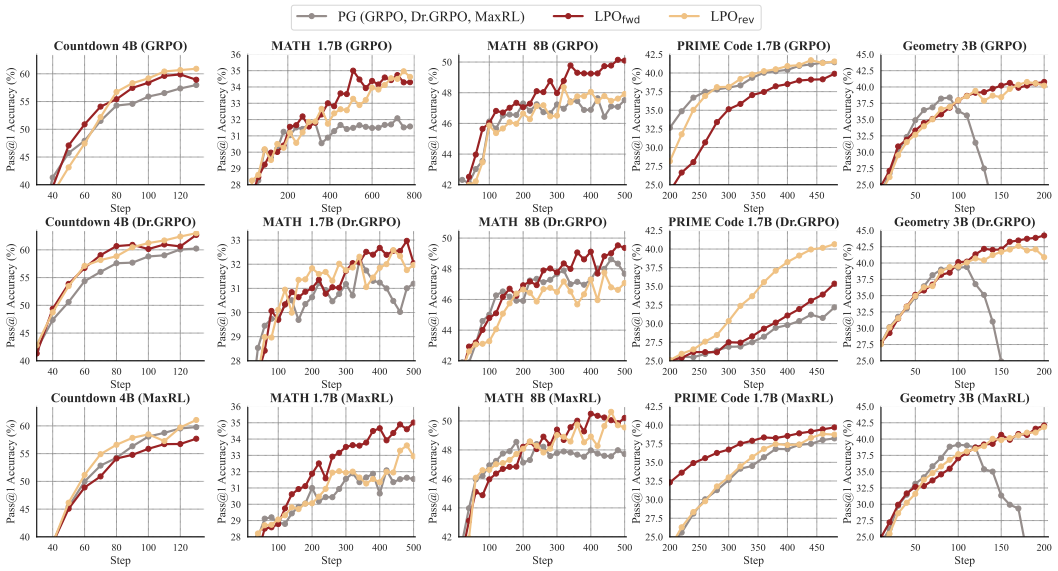


Figure 3: Training curves of Pass@1 accuracy. Two LPO variants (LPO_{fwd}, LPO_{rev}) are evaluated against group-based PG baselines (GRPO, Dr.GRPO, MaxRL, shown from top to bottom) across various LLM backbones and reasoning tasks with corresponding temperature designs.

following Qu et al. [2025b], Gao et al. [2025]: AIME24, AIME25, AMC23, MATH500 [Lightman et al., 2023], Minerva Math [Lewkowycz et al., 2022], and OlympiadBench [He et al., 2024]. In Appendix E.1, we scale to Qwen3-14B-Base on the larger Polaris dataset [An et al., 2025].

Programming. We train and evaluate Qwen3-1.7B-Base on the respective training and test splits of the PRIME code dataset [Cui et al., 2025].

Multi-Modal Geometry. Geometry problems require multi-modal understanding and reasoning. We train Qwen2.5-VL-3B-Instruct [Bai et al., 2025] on the training split of the Geometry3k dataset [Lu et al., 2021, Hiyouga, 2025] and evaluate it on the test split.

Baselines and LPO Variants. We compare against three representative group-based policy gradient (**PG**) methods with varied target temperature designs: **GRPO** ($\tau = \sigma_G$)**, Dr.GRPO** ($\tau = 1$)**, and MaxRL** ($\tau = \mu_G$). To ensure a rigorous apples-to-apples comparison, we isolate the effect of the gradient formulation from temperature scaling by implementing LPO variants for each baseline. Specifically, we develop forward ($\mathbf{LPO_{fwd}}$) and reverse KL ($\mathbf{LPO_{rev}}$) versions that use the exact same temperature τ as their corresponding PG counterpart. The paired evaluation ensures that any performance differences are attributable to explicit listwise projection rather than temperature tuning. We implement baselines and LPO with the verl framework [Sheng et al., 2024].

Additional implementation details are provided in Appendix D, together with extended experimental results in Appendix E and prompt examples in Appendix F.

5.2 Training Performance

Performance gains. Under paired temperature configurations, LPO consistently outperforms group-based PG baselines. For Pass@1 accuracy in Fig. 3, both LPO variants demonstrate efficient and improved training performance, exceeding their corresponding PG baselines in nearly all settings (13/15 for LPO_{fwd} and 13/15 for LPO_{rev}). This advantage also extends to Pass@k evaluations in Fig. 4, where both LPO variants continue to surpass the implicit PG methods (15/15 for LPO_{fwd} and 11/15 for LPO_{rev}). Together, these consistent gains suggest that replacing first-order advantage approximations with exact listwise projection on the response simplex offers a promising paradigm for improving the training efficiency and performance of RLVR.

Projection divergence effects. Comparing the two variants reveals an empirical distinction: LPO_{fwd} outperforms LPO_{rev} in 13/15 scenarios for Pass@k. This observation aligns well with the expectation: the mode-coverage property inherent to forward-KL actively preserves reasoning diversity for a broader distribution of valid solution paths. More broadly, this highlights the flexibility of the

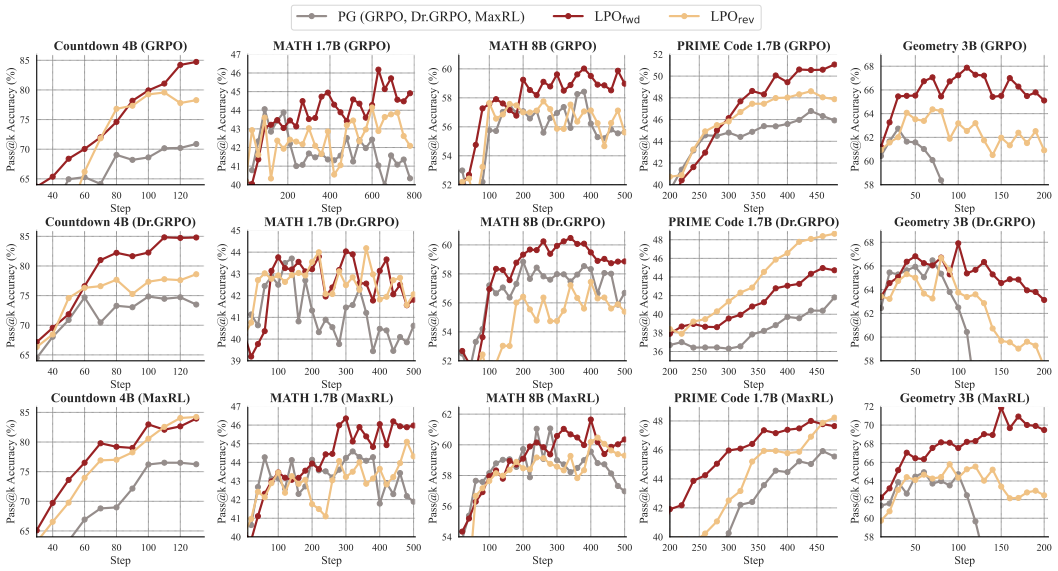


Figure 4: Pass@k training curves. LPO variants (LPO_{fwd}, LPO_{rev}) are evaluated against group-based PG baselines (GRPO, Dr.GRPO, MaxRL, shown from top to bottom) across various LLMs and tasks under paired temperature settings. Specific k configurations are detailed per benchmark.

decoupled target-projection framework, suggesting that exploring alternative projection divergences could unlock further unique optimization properties.

Robustness across temperature parameterizations. We observe that the optimal implicit temperature strategy τ is highly task-dependent, with no single design consistently dominating across all benchmarks. Despite this task-varying behavior, LPO delivers stable performance gains under all tested τ designs. This indicates that exact listwise projection provides a robust optimization mechanism, yielding benefits that are largely orthogonal to the underlying temperature heuristic.

5.3 Training Dynamics

To better understand the underlying optimization behaviors and validate our theoretical analysis, we track key training metrics: response entropy, gradient norm, and response length.

Response entropy and exploration preservation. As shown in Fig. 5 (top), both LPO variants generally maintain higher response entropy than PG baselines. This corresponds to the projection properties: LPO_{rev} corresponds to a maximum-entropy objective, while LPO_{fwd} exhibits mode-covering behavior. This sustained diversity directly explains the robust Pass@k improvements, positioning listwise projection as a principled remedy for the entropy collapse in RLVR.

Gradient norms and optimization stability. Fig. 5 (middle) reveals that LPO variants exhibit lower and more stable gradient norms compared to group-based PG methods. This empirical stability is consistent with Corollary 1: LPO's exact projection on the response simplex yields controlled gradient coefficients, leading to stable optimization dynamics.

Response length and reasoning behaviors. Fig. 5 (bottom) shows that LPO tends to generate longer responses than PG. As increased length often correlates with more detailed reasoning chains [Yu et al., 2025], this is consistent with LPO encouraging more extensive exploration. LPO_{fwd}'s maximum length aligns with its mode-covering property, which promotes diverse reasoning paths.

5.4 Additional Analysis

5.4.1 Listwise vs. Pointwise Projection

To highlight the contribution of the listwise projection, we ablate the listwise policy distribution in Eq. (4) while keeping the target in Eq. (8) unchanged. This recovers the **pointwise projection** with forward KL [Peters et al., 2010, Abdolmaleki et al., 2018, Peng et al., 2019], defined as $\mathcal{L}_{\text{point}} = -\sum_k w_k^* \log \pi_\theta(y_k|x)$. As shown in Fig. 6, this pointwise variant suffers a severe performance drop.

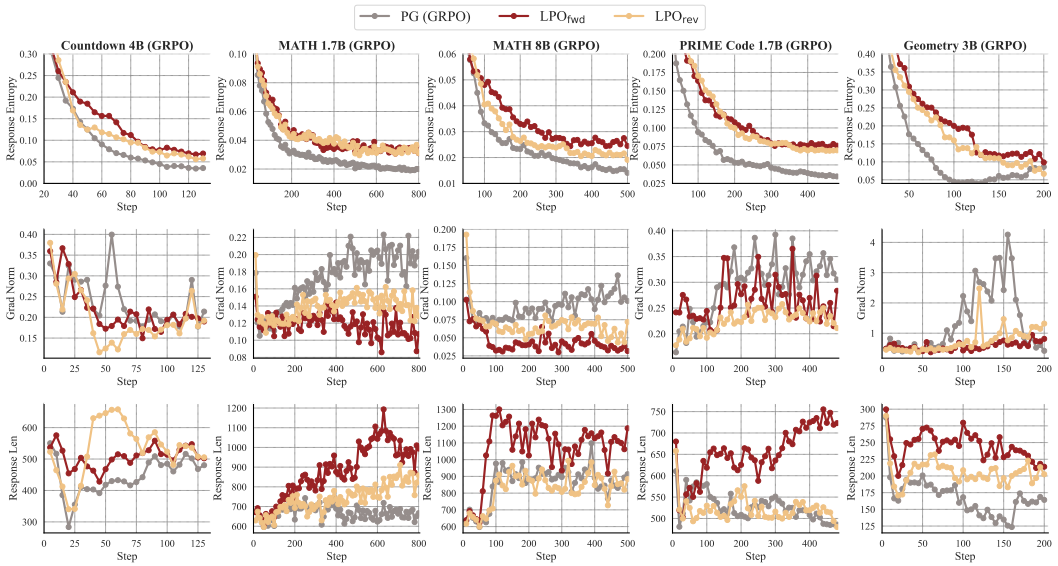


Figure 5: Training dynamics of LPO variants and GRPO. Rows from top to bottom respectively show the curves of response entropy, gradient norms, and response lengths.

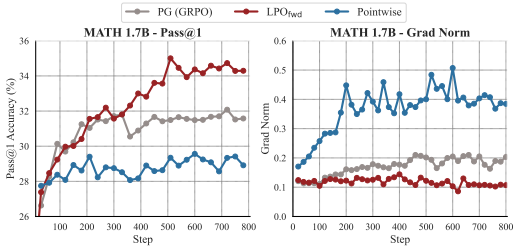


Figure 6: Ablation comparing listwise LPO with pointwise projection and GRPO baselines on MATH (Qwen3-1.7B-Base).

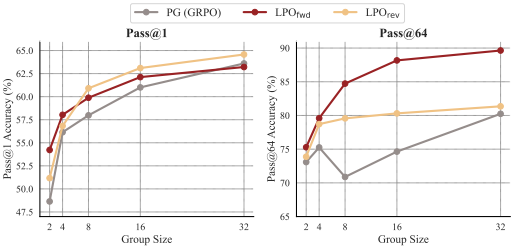


Figure 7: Effect of varying group sizes $K \in \{2, 4, 8, 16, 32\}$ on Countdown.

This failure stems from the lack of a coupled competitive mechanism across responses in pointwise updates, resulting in unstable optimization. Conversely, both group-based PG and LPO intrinsically employ a built-in control variate that stabilizes training. These results suggest that our performance gains stem not merely from the target design, but from successfully marrying exact target fitting with the crucial structural variance reduction provided by the listwise projection. Detailed properties of the two projections are deferred to Appendix C.4.

5.4.2 Effect of Group Size K

We investigate the impact of the sampled group size K on Countdown. As shown in Fig. 7, across the tested group sizes ($K \in \{2, 4, 8, 16, 32\}$), both LPO variants remain highly competitive with GRPO, with advantages being particularly pronounced at smaller group sizes. This suggests that explicit listwise projection improves sample efficiency, which stabilizes updates under limited samples. Furthermore, LPO variants exhibit distinct scaling behaviors that validate their theoretical properties: LPO_{rev} achieves stronger Pass@1 performance, while LPO_{fwd} scales exceptionally well on Pass@64, supporting its mode-coverage property which structurally preserves reasoning diversity.

5.4.3 Generalization across LLM Families

To evaluate the generalizability of LPO, we conduct experiments across four prominent LLM families: Qwen, DeepSeek, Mistral, and Llama. As illustrated in Fig. 11 in Appendix E.3, LPO consistently delivers performance gains on the Countdown task, regardless of the underlying model architecture or training paradigm. This consistent improvement across diverse backbones suggests that LPO is not sensitive to a specific model architecture, but rather benefits from the fundamental robustness of

the listwise projection framework. These results indicate that LPO can serve as a model-agnostic approach for improving reasoning performance in RLVR.

6 Conclusion

This work introduces a unified geometric framework for deep insight into group-based RLVR of LLMs. We show that existing policy gradient methods act as approximate target-projections on the response simplex and present LPO to perform this projection explicitly. LPO benefits from directly optimizing on the simplex, which improves optimization stability and yields monotonic performance improvements. Moreover, the decoupled target-projection perspective opens up a flexible design space for developing rich and diverse optimization methods for RLVR of LLMs.

Limitations and future work. Our current formulation primarily focuses on sequence-level projection within outcome reward settings. Future research will explore step-level listwise projections and investigate broader divergences to fully unlock the potential of the decoupled framework.

Acknowledgments and Disclosure of Funding

This work was supported by the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China JYB2025XDXM503 and National Natural Science Foundation of China 62495092. Dr. Qi Wang acknowledges support from the National Natural Science Foundation of China under the grant number 62306326.

References

Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, and Martin Riedmiller. Maximum a posteriori policy optimisation. *arXiv preprint arXiv:1806.06920*, 2018.

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12248–12267, 2024.

Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276, 1998.

Chenxin An, Zhihui Xie, Xiaonan Li, Lei Li, Jun Zhang, Shansan Gong, Ming Zhong, Jingjing Xu, Xipeng Qiu, Mingxuan Wang, and Lingpeng Kong. Polaris: A post-training recipe for scaling reinforcement learning on advanced reasoning models, 2025. URL https://hkunlp.github.io/blog/2025/Polaris.

GX-Chen Anthony, Jatin Prakash, Rob Fergus, and Rajesh Ranganath. Reverse-kl reinforcement learning can sample from multiple diverse modes. In *First Workshop on Foundations of Reasoning in Language Models*.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136, 2007.

Aili Chen, Aonian Li, Bangwei Gong, Binyang Jiang, Bo Fei, Bo Yang, Boji Shan, Changqing Yu, Chao Wang, Cheng Zhu, et al. Minimax-m1: Scaling test-time compute efficiently with lightning attention. *arXiv preprint arXiv:2506.13585*, 2025.

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. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, et al. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025.

Peter Dayan and Geoffrey E Hinton. Using expectation-maximization for reinforcement learning. *Neural Computation*, 9(2):271–278, 1997.

Zhaolin Gao, Joongwon Kim, Wen Sun, Thorsten Joachims, Sid Wang, Richard Yuanzhe Pang, and Liang Tan. Prompt curriculum learning for efficient llm post-training. *arXiv preprint arXiv:2510.01135*, 2025.

Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. A theory of regularized markov decision processes. In *International conference on machine learning*, pages 2160–2169. PMLR, 2019.

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

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. Pmlr, 2018.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*, 2024.

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

Hiyouga. Geometry3K: A large-scale multi-modal geometry reasoning dataset. https://huggingface.co/datasets/hiyouga/geometry3k, 2025.

Jian Hu. Reinforce++: A simple and efficient approach for aligning large language models. *arXiv e-prints*, pages arXiv–2501, 2025.

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

Harold Jeffreys. An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 186(1007):453–461, 1946.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

Jean Kaddour. Target policy optimization. *arXiv preprint arXiv:2604.06159*, 2026a.

Jean Kaddour. Target policy optimization, 2026b. URL https://arxiv.org/abs/2604.06159.

Sham M Kakade. A natural policy gradient. *Advances in neural information processing systems*, 14, 2001.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Solomon Kullback. Kullback-leibler divergence. *Encyclopedia of Machine Learning*, pages 581–583, 1951.

Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857, 2022.

Long Li, Zhijian Zhou, Jiaran Hao, Jason Klein Liu, Yanting Miao, Wei Pang, Xiaoyu Tan, Wei Chu, Zhe Wang, Shirui Pan, et al. The choice of divergence: A neglected key to mitigating diversity collapse in reinforcement learning with verifiable reward. *arXiv preprint arXiv:2509.07430*, 2025.

Ziniu Li, Tian Xu, Yushun Zhang, Zhihang Lin, Yang Yu, Ruoyu Sun, and Zhi-Quan Luo. Remax: A simple, effective, and efficient reinforcement learning method for aligning large language models. *arXiv preprint arXiv:2310.10505*, 2023.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2023.

Tianqi Liu, Zhen Qin, Junru Wu, Jiaming Shen, Misha Khalman, Rishabh Joshi, Yao Zhao, Mohammad Saleh, Simon Baumgartner, Jialu Liu, Peter J. Liu, and Xuanhui Wang. Lipo: Listwise preference optimization through learning-to-rank, 2025a. URL https://arxiv.org/abs/2402.01878.

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

Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. Inter-gps: Interpretable geometry problem solving with formal language and symbolic reasoning. In *The Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021)*, 2021.

R Duncan Luce et al. *Individual choice behavior*, volume 4. Wiley New York, 1959.

Michael Luo, Sijun Tan, Roy Huang, Ameen Patel, Alpay Ariyak, Qingyang Wu, Xiaoxiang Shi, Rachel Xin, Colin Cai, Maurice Weber, et al. Deepcoder: A fully open-source 14b coder at o3-mini level. *Notion Blog*, 2025.

Yixiu Mao, Yun Qu, Qi Wang, Heming Zou, and Xiangyang Ji. Dynamics-predictive sampling for active rl finetuning of large reasoning models. In *International Conference on Learning Representations*, volume 2026, pages 25242–25270, 2026a.

Yixiu Mao, Yun Qu, Qi Wang, Heming Zou, and Xiangyang Ji. Rlvr without ineffective samples: Group prioritized off-policy optimization for llm reasoning. *arXiv preprint arXiv:2606.01281*, 2026b.

Youssef Mroueh. Reinforcement learning with verifiable rewards: Grpo's effective loss, dynamics, and success amplification. *arXiv preprint arXiv:2503.06639*, 2025.

Radford M Neal and Geoffrey E Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pages 355–368. Springer, 1998.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Jiayi Pan, Junjie Zhang, Xingyao Wang, Lifan Yuan, Hao Peng, and Alane Suhr. Tinyzero. https://github.com/Jiayi-Pan/TinyZero, 2025. Accessed: 2025-01-24.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

Jan Peters, Katharina Mulling, and Yasemin Altun. Relative entropy policy search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, pages 1607–1612, 2010.

Robin L Plackett. The analysis of permutations. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 24(2):193–202, 1975.

Yun Qu, Boyuan Wang, Yuhang Jiang, Jianzhun Shao, Yixiu Mao, Cheems Wang, Chang Liu, and Xiangyang Ji. Choices are more important than efforts: Llm enables efficient multi-agent exploration. *arXiv preprint arXiv:2410.02511*, 2024.

Yun Qu, Yuhang Jiang, Boyuan Wang, Yixiu Mao, Cheems Wang, Chang Liu, and Xiangyang Ji. Latent reward: Llm-empowered credit assignment in episodic reinforcement learning. *AAAI*, 2025a.

Yun Qu, Qi Wang, Yixiu Mao, Vincent Tao Hu, Björn Ommer, and Xiangyang Ji. Can prompt difficulty be online predicted for accelerating rl finetuning of reasoning models? *arXiv preprint arXiv:2507.04632*, 2025b.

Yun Qu, Qi Cheems Wang, Yixiu Mao, Yiqin Lv, and Xiangyang Ji. Fast and robust: Task sampling with posterior and diversity synergies for adaptive decision-makers in randomized environments. *arXiv preprint arXiv:2504.19139*, 2025c.

Yun Qu, Qi Wang, Yixiu Mao, Heming Zou, Yuhang Jiang, Weijie Liu, Clive Bai, Kai Yang, Yangkun Chen, Saiyong Yang, et al. Small generalizable prompt predictive models can steer efficient rl post-training of large reasoning models. *arXiv preprint arXiv:2602.01970*, 2026.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2024. URL https://arxiv.org/abs/2305.18290.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024.

John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust region policy optimization, 2017a. URL https://arxiv.org/abs/1502.05477.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017b.

Jianzhun Shao, Yun Qu, Chen Chen, Hongchang Zhang, and Xiangyang Ji. Counterfactual conservative q learning for offline multi-agent reinforcement learning. *NeurIPS*, 36, 2024a.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024b.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024.

Yao Shu, Chenxing Wei, Hongbin Lin, Shuang Qiu, and Hui Xiong. Reference-sampled boltzmann projection for kl-regularized rlvr: Target-matched weighted sft, finite one-shot gaps, and policy mirror descent. *arXiv preprint arXiv:2605.02469*, 2026.

H Francis Song, Abbas Abdolmaleki, Jost Tobias Springenberg, Aidan Clark, Hubert Soyer, Jack W Rae, Seb Noury, Arun Ahuja, Siqi Liu, Dhruva Tirumala, et al. V-mpo: On-policy maximum a posteriori policy optimization for discrete and continuous control. *arXiv preprint arXiv:1909.12238*, 2019.

Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3 (1):9–44, 1988.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

Fahim Tajwar, Guanning Zeng, Yueer Zhou, Yuda Song, Daman Arora, Yiding Jiang, Jeff Schneider, Ruslan Salakhutdinov, Haiwen Feng, and Andrea Zanette. Maximum likelihood reinforcement learning. *arXiv preprint arXiv:2602.02710*, 2026.

Manan Tomar, Lior Shani, Yonathan Efroni, and Mohammad Ghavamzadeh. Mirror descent policy optimization. *arXiv preprint arXiv:2005.09814*, 2020.

Milan Vojnovic and Se-Young Yun. What is the alignment objective of grpo? *arXiv preprint arXiv:2502.18548*, 2025.

Chaoqi Wang, Yibo Jiang, Chenghao Yang, Han Liu, and Yuxin Chen. Beyond reverse kl: Generalizing direct preference optimization with diverse divergence constraints. *arXiv preprint arXiv:2309.16240*, 2023.

Qi Cheems Wang, Zehao Xiao, Yixiu Mao, Yun Qu, Jiayi Shen, Yiqin Lv, and Xiangyang Ji. Model predictive task sampling for efficient and robust adaptation. *Nature Communications*, 2026.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37: 95266–95290, 2024.

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

Wei Yang, Jiacheng Pang, Shixuan Li, Paul Bogdan, Stephen Tu, and Jesse Thomason. Maestro: Learning to collaborate via conditional listwise policy optimization for multi-agent llms. *arXiv preprint arXiv:2511.06134*, 2025b.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

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

Xinyu Zhu, Mengzhou Xia, Zhepei Wei, Wei-Lin Chen, Danqi Chen, and Yu Meng. The surprising effectiveness of negative reinforcement in llm reasoning. *Advances in Neural Information Processing Systems*, 38:126546–126573, 2026.

Xuekai Zhu, Daixuan Cheng, Dinghuai Zhang, Hengli Li, Kaiyan Zhang, Che Jiang, Youbang Sun, Ermo Hua, Yuxin Zuo, Xingtai Lv, et al. Flowrl: Matching reward distributions for llm reasoning. *arXiv preprint arXiv:2509.15207*, 2025.

Brian D Ziebart. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. Carnegie Mellon University, 2010.

Heming Zou, Qi Wang, Yixiu Mao, Yun Qu, Yuyi Guo, Zeyuan Liu, Chenglin Li, Weijie Liu, Saiyong Yang, and Xiangyang Ji. Tree-aware posterior distillation for long-horizon agentic reinforcement learning. *arXiv preprint arXiv:2610.23181*, 2026a.

Heming Zou, Qi Wang, Yun Qu, Yuhang Jiang, Lizhou Cai, Yixiu Mao, Ru Peng, Xin Xu, Weijie Liu, Kai Yang, et al. Trace: A unified rollout budget allocation framework for efficient agentic reinforcement learning. *arXiv preprint arXiv:2606.11119*, 2026b.

Appendix Overview

This appendix provides supplementary proofs, conceptual discussions, and experimental details supporting the main text. It is organized as follows:

- **Appendix A (Related Works):** reviews literature on RLVR, RL as probabilistic inference, and listwise formulations.
- **Appendix B (Proofs):** provides detailed mathematical derivations for all theoretical claims.
- **Appendix C (Additional Discussions):** expands on the framework's conceptual and practical scope. It unifies existing group-based RLVR algorithms, compares listwise projection with pointwise and preference optimization, explores future extensions.
- **Appendix D (Implementation Details):** outlines the experimental setup, including tasks, LLM backbones, and training details.
- **Appendix E (Extended Experimental Results):** reports further empirical findings, including scalability validation, on-policy optimization, extended training dynamics, and generalization across diverse LLM families.
- **Appendix F (Data Examples):** presents representative data examples used across the evaluated reasoning tasks.

A Related Works

Reinforcement learning with verifiable rewards. The alignment and reasoning capabilities of LLMs have been significantly advanced by RL [Ouyang et al., 2022, Bai et al., 2022, Zou et al., 2026b], initially dominated by PPO [Schulman et al., 2017b] with a learned value model. The emergence of RLVR for reasoning tasks [Jaech et al., 2024, Guo et al., 2025] has driven a paradigm shift toward critic-free, group-based policy gradient methods [Shao et al., 2024b, Ahmadian et al., 2024, Li et al., 2023], which sample multiple responses per prompt and derive advantages entirely from within-group reward statistics. A subsequent line of work has refined this paradigm, introducing novel advantage normalization [Liu et al., 2025b, Hu, 2025, Tajwar et al., 2026], trust-region mechanics [Yu et al., 2025, Chen et al., 2025], and sequence-level scaling [Zheng et al., 2025]. Recent theoretical works have sought to uncover the underlying mechanics of these methods [Mroueh, 2025, Vojnovic and Yun, 2025, Zhu et al., 2026]. Our LPO framework provides a unifying perspective by revealing that major group-based methods share the same target-projection geometry. Concurrently, FlowRL [Zhu et al., 2025] minimizes reverse KL against a Gibbs target approximated by a learned partition function network, while contemporaneous TPO [Kaddour, 2026b] adopts cross-entropy on tilted simplex targets. Differently, LPO contributes a unifying analytical Target-Projection framework that recovers existing group-based methods, and admits multiple divergences with provably satisfying properties.

RL as probabilistic inference. The idea of constructing a reward-weighted target distribution and projecting the policy toward it has deep roots in the RL-as-inference literature, which casts control as inference under a KL-regularized objective [Dayan and Hinton, 1997, Ziebart, 2010, Levine, 2018, Geist et al., 2019, Shao et al., 2024a]. This perspective gives rise to a natural trust-region structure [Amari, 1998, Kakade, 2001, Schulman et al., 2017a] and underlies a wide range of practical algorithms [Peters et al., 2010, Abdolmaleki et al., 2018, Song et al., 2019, Peng et al., 2019, Haarnoja et al., 2018, Tomar et al., 2020]. However, these methods typically operate in continuous action spaces and resort to *pointwise* projections, i.e., $-\sum_k w_k^* \log \pi_\theta(y_k)$. In contrast, sampled responses in LLM form a finite response simplex, where normalization is exact and the partition function reduces to a finite sum over samples. This structure enables listwise projection on the simplex, as exploited by LPO, which couples all responses through shared normalization and inherits satisfying gradients (Appendix C.4).

Listwise formulation. Listwise formulation has a long history in classical choice and learning-to-rank models [Luce et al., 1959, Plackett, 1975, Cao et al., 2007], where a distribution over candidate sets or permutations is modeled or optimized. Recent LLM alignment methods, such as DPO [Rafailov et al., 2024] and its extensions [Liu et al., 2025a], adopt pairwise or listwise preference structures to model relative comparisons among responses. Listwise structures have also

been employed in multi-agent LLM collaboration [Yang et al., 2025b]. Our approach operates in the standard RLVR setting for LLM post-training and explicitly constructs a target distribution based on verifiable rewards on the response simplex, followed by direct projection onto it.

B Proofs

B.1 KL Gradient Derivations

We derive the gradients of the forward and reverse KL divergences stated in Section 4.2. For both derivations, we recall from Eq. (4) that the logits are defined as $s_{\theta,k} = \log \pi_\theta(y_k|x) - \log \pi_b(y_k|x)$. Since the behavior policy π_b is frozen, the gradient of the logit with respect to the parameters is simply $\nabla_\theta s_{\theta,k} = \nabla_\theta \log \pi_\theta(y_k|x)$.

Forward KL: $D_{\mathrm{KL}}(w^*\|P_\theta)$. By definition, $D_{\mathrm{KL}}(w^*\|P_\theta) = -\sum_{k=1}^K w_k^* \log P_{\theta,k} - H(w^*)$, where $H(w^*)$ is the entropy of w^*, which is constant with respect to θ. Using the fact that $\log P_{\theta,k} = s_{\theta,k} - \log \sum_{j=1}^K e^{s_{\theta,j}}$, the Jacobian of the log-softmax is given by

$$\nabla_\theta \log P_{\theta,k} = \nabla_\theta s_{\theta,k} - \sum_{j=1}^K P_{\theta,j} \nabla_\theta s_{\theta,j}. \tag{12}$$

Substituting this into the gradient of the forward KL divergence, we obtain:

$$\begin{aligned}\nabla_\theta D_{\mathrm{KL}}(w^*\|P_\theta) &= -\sum_{k=1}^K w_k^* \left(\nabla_\theta s_{\theta,k} - \sum_{j=1}^K P_{\theta,j} \nabla_\theta s_{\theta,j} \right) \\ &= -\sum_{k=1}^K w_k^* \nabla_\theta s_{\theta,k} + \left(\sum_{k=1}^K w_k^* \right) \sum_{j=1}^K P_{\theta,j} \nabla_\theta s_{\theta,j}.\end{aligned}$$

Since w^* is a valid probability distribution ($\sum_{k=1}^K w_k^* = 1$), the second term simplifies. Reindexing the summation and substituting $\nabla_\theta s_{\theta,k} = \nabla_\theta \log \pi_\theta(y_k|x)$, we get:

$$\nabla_\theta D_{\mathrm{KL}}(w^*\|P_\theta) = \sum_{k=1}^K (P_{\theta,k} - w_k^*) \nabla_\theta \log \pi_\theta(y_k|x). \tag{13}$$

Reverse KL: $D_{\mathrm{KL}}(P_\theta\|w^*)$. Write the reverse KL as $D_{\mathrm{KL}}(P_\theta\|w^*) = \sum_{k=1}^K P_{\theta,k} \log(P_{\theta,k}/w_k^*)$. We first compute the partial derivative with respect to a single logit $s_{\theta,j}$. Using the standard softmax Jacobian $\partial P_{\theta,k}/\partial s_{\theta,j} = P_{\theta,k}(\delta_{kj} - P_{\theta,j})$ and applying the product rule, we have:

$$\begin{aligned}\frac{\partial}{\partial s_{\theta,j}} D_{\mathrm{KL}}(P_\theta\|w^*) &= \sum_{k=1}^K P_{\theta,k}(\delta_{kj} - P_{\theta,j}) \big[\log(P_{\theta,k}/w_k^*) + 1\big] \\ &= P_{\theta,j}\big[\log(P_{\theta,j}/w_j^*) + 1\big] - P_{\theta,j} \sum_{k=1}^K P_{\theta,k}\big[\log(P_{\theta,k}/w_k^*) + 1\big] \\ &= P_{\theta,j} \log \frac{P_{\theta,j}}{w_j^*} + P_{\theta,j} - P_{\theta,j} D_{\mathrm{KL}}(P_\theta\|w^*) - P_{\theta,j}(1) \\ &= P_{\theta,j}\Big[\log \frac{P_{\theta,j}}{w_j^*} - D_{\mathrm{KL}}(P_\theta\|w^*)\Big].\end{aligned}$$

Applying the multivariate chain rule $\nabla_\theta D_{\mathrm{KL}} = \sum_{j=1}^K \frac{\partial D_{\mathrm{KL}}}{\partial s_{\theta,j}} \nabla_\theta s_{\theta,j}$, we arrive at the full gradient:

$$\nabla_\theta\, D_{\mathrm{KL}}(P_\theta\|w^*) = \sum_{j=1}^K P_{\theta,j}\Big[\log \frac{P_{\theta,j}}{w_j^*} - D_{\mathrm{KL}}(P_\theta\|w^*)\Big] \nabla_\theta \log \pi_\theta(y_j|x). \quad \square \tag{14}$$

Logit-gap simplification for reverse KL. The per-logit coefficient for the reverse KL gradient, $c_k^{\text{rev}} = \frac{\partial}{\partial s_{\theta,k}} D_{\text{KL}}(P_\theta \| w^*)$, can be elegantly simplified when written in terms of the logit gap $d_k = s_{\theta,k} - \phi_k$, where ϕ_k is the target logit from Eq. (8).

Express both probabilities explicitly with their partition functions: $P_{\theta,k} = \exp(s_{\theta,k})/Z_s$ and $w_k^* = \exp(\phi_k)/Z_\phi$. The log-ratio becomes:

$$\log \frac{P_{\theta,k}}{w_k^*} = (s_{\theta,k} - \log Z_s) - (\phi_k - \log Z_\phi) = (s_{\theta,k} - \phi_k) - (\log Z_s - \log Z_\phi) = d_k - c_s, \quad (15)$$

where $c_s = \log Z_s - \log Z_\phi$ is strictly constant across all k. Consequently, the KL divergence can be written in terms of the expected gap:

$$D_{\text{KL}}(P_\theta \| w^*) = \sum_{k=1}^{K} P_{\theta,k}(d_k - c_s) = \bar{d} - c_s, \quad (16)$$

where $\bar{d} = \sum_{k=1}^{K} P_{\theta,k} d_k$. Substituting these back into the coefficient c_k^{rev}, the constant c_s perfectly cancels out:

$$c_k^{\text{rev}} = P_{\theta,k}\big[(d_k - c_s) - (\bar{d} - c_s)\big] = P_{\theta,k}(d_k - \bar{d}). \quad (17)$$

This reveals that the reverse KL gradient admits a baseline-subtracted form, where the baseline corresponds to the expected logit gap under the current policy P_θ.

B.2 Proof of Proposition 1

Proposition 1 (Group-based policy gradient as reverse KL at on-policy)**.** *Let $A \in \mathbb{R}^K$ be a zero-mean advantage vector, i.e., $\sum_{k=1}^{K} A_k = 0$, and let $w^* = \text{softmax}(A)$. At the on-policy point ($\pi_\theta = \pi_b$), the policy gradient in Eq. (3) equals the negative gradient of the reverse KL divergence D_{KL}:*

$$g_{\text{PG}} = \frac{1}{K} \sum_{k=1}^{K} A_k \nabla_\theta \log \pi_\theta(y_k|x) = -\nabla_\theta D_{\text{KL}}(P_\theta \| w^*)\Big|_{\pi_\theta = \pi_b}. \quad (5)$$

Proof. By the logit-gap simplification derived in Eq. (17), the reverse KL gradient is fully characterized by the per-logit coefficients:

$$c_k^{\text{rev}} = P_{\theta,k}(d_k - \bar{d}), \quad (18)$$

where $d_k = s_{\theta,k} - A_k$ and $\bar{d} = \sum_{k=1}^{K} P_{\theta,k} d_k$. We directly evaluate this coefficient at the on-policy point $\pi_\theta = \pi_b$.

At the on-policy point, the logit offsets vanish ($s_{\theta,k} = 0$ for all k), which yields a uniform probability distribution over the generated list: $P_{\theta,k} = \text{softmax}(\mathbf{0})_k = 1/K$. Consequently, the logit gap simplifies to $d_k = -A_k$.

Applying the zero-mean advantage assumption $\sum_{k=1}^{K} A_k = 0$, the expected gap identically vanishes:

$$\bar{d} = \frac{1}{K} \sum_{k=1}^{K} (-A_k) = 0. \quad (19)$$

Substituting these on-policy values back into the coefficient yields:

$$c_k^{\text{rev}}\Big|_{\pi_\theta = \pi_b} = \frac{1}{K}(-A_k - 0) = -\frac{A_k}{K}. \quad (20)$$

Recall that the standard policy gradient in Eq. (3) can be expressed as $g_{\text{PG}} = \sum_k c_k^{\text{PG}} \nabla_\theta \log \pi_\theta(y_k|x)$ with coefficients $c_k^{\text{PG}} = A_k/K$. Comparing the coefficients, we immediately obtain $c_k^{\text{PG}} = -c_k^{\text{rev}}|_{\pi_\theta = \pi_b}$, which proves that

$$g_{\text{PG}} = -\nabla_\theta D_{\text{KL}}(P_\theta \| w^*)\big|_{\pi_\theta = \pi_b}. \quad (21)$$

confirming that the policy gradient step is a gradient descent step on the reverse KL divergence at the on-policy point.

The centering assumption $\sum_k A_k = 0$ is without loss of generality: by the shift-invariance of softmax, replacing A with $A - \bar{A}$ does not change the target $w^* = \mathrm{softmax}(A)$.

Finally, we clarify that the zero-mean assumption, i.e., $\sum_k A_k = 0$, is not a restrictive algorithmic requirement, but rather a natural reflection of the listwise projection's intrinsic mechanics. Due to the shift-invariance of the target softmax, any prompt-level scalar baseline applied uniformly to the group's rewards, e.g., the greedy baseline in ReMax [Li et al., 2023], is mathematically absorbed and nullified. The zero-sum constraint of the local simplex inherently induces a dynamically weighted mean-centering control variate $(d_k - \bar{d})$ during the gradient computation. At the on-policy point, this natively recovers the arithmetic zero-mean counterpart $(A_k - \bar{A})$. Thus, assuming centered advantages simply aligns our notation with the framework's built-in behavior at the exact point of equivalence.

□

Off-policy approximation error. Proposition 1 establishes exact equality at $\pi_\theta = \pi_b$. We now quantify the discrepancy off-policy, incorporating the importance sampling ratio $r_k = \pi_\theta(y_k|x)/\pi_b(y_k|x)$ standard in practical PG methods without clipping.

Let $s_{\theta,k} = \log r_k$, $P_\theta = \mathrm{softmax}(s_\theta)$, and $\bar{\delta} = \max_k |r_k - 1|$. Both the policy gradient and the reverse KL gradient can be written as $\sum_k c_k \nabla_\theta \log \pi_\theta(y_k|x)$ with respective coefficients

$$c_k^{\mathrm{PG}} = \frac{r_k A_k}{K}, \qquad c_k^{\mathrm{revKL}} = -P_{\theta,k}(d_k - \bar{d}), \tag{22}$$

where $d_k = s_{\theta,k} - A_k$ is the logit gap from Eq. (17) and $\bar{d} = \sum_k P_{\theta,k} d_k$. The per-coefficient discrepancy is

$$\Delta_k \;=\; c_k^{\mathrm{PG}} - c_k^{\mathrm{revKL}} \;=\; \frac{r_k A_k}{K} + P_{\theta,k}(d_k - \bar{d}), \tag{23}$$

which vanishes identically at on-policy point ($\pi_\theta = \pi_b$).

We analyze the local regime where $\bar{\delta} < 1/2$, under which we have $r_k \in [1/2, 3/2]$, and thus $\|s_\theta\|_\infty \le 2\bar{\delta}$.

A first-order Taylor expansion of $P_{\theta,k} = \mathrm{softmax}(s_\theta)_k$ and $r_k = \exp(s_{\theta,k})$ around the zero vector $s_\theta = \mathbf{0}$ gives

$$P_{\theta,k} = \frac{1}{K} + \frac{s_{\theta,k} - \bar{s}_\theta}{K} + O\!\left(\frac{\|s_\theta\|_\infty^2}{K}\right), \quad r_k = 1 + s_{\theta,k} + O(\|s_\theta\|_\infty^2), \tag{24}$$

where $\bar{s}_\theta = \frac{1}{K}\sum_k s_{\theta,k}$. Using the zero-mean advantage assumption $\sum_k A_k = 0$, the first-order expansion of $\bar{d} = \sum_k P_{\theta,k}(s_{\theta,k} - A_k)$ yields

$$\bar{d} = \bar{s}_\theta - \frac{1}{K}\sum_m s_{\theta,m} A_m + O(\bar{\delta}^2). \tag{25}$$

Collecting terms:

$$\Delta_k = \frac{1}{K}\Big[(s_{\theta,k} - \bar{s}_\theta) + \bar{s}_\theta A_k + \frac{1}{K}\textstyle\sum_m s_{\theta,m} A_m\Big] + O\!\left(\tfrac{\bar{\delta}^2(1+\|A\|_\infty)}{K}\right). \tag{26}$$

Bounding the three terms via $|s_{\theta,k} - \bar{s}_\theta| \le 2\|s_\theta\|_\infty \le 4\bar{\delta}$, and symmetrically for the advantage terms $|\bar{s}_\theta A_k| \le 2\bar{\delta}\|A\|_\infty$ and $|\frac{1}{K}\sum_m s_{\theta,m} A_m| \le \|s_\theta\|_\infty \|A\|_\infty \le 2\bar{\delta}\|A\|_\infty$:

$$|\Delta_k| \;\le\; \frac{C\,\bar{\delta}\,(1 + \|A\|_\infty)}{K} \tag{27}$$

for a universal constant $C > 0$, to leading order in $\bar{\delta}$. By the triangle inequality, the parameter-space gradient discrepancy satisfies

$$\|g_{\mathrm{PG}} - g_{\mathrm{revKL}}\| \;\le\; \textstyle\sum_k |\Delta_k|\, G_{\max} \;\le\; C'\bar{\delta}(1 + \|A\|_\infty)\, G_{\max}, \tag{28}$$

where $G_{\max} = \max_k \|\nabla_\theta \log \pi_\theta(y_k|x)\|$. The error is linear in the off-policy drift $\bar{\delta}$ and vanishes at the on-policy point, confirming that the policy gradient approximates reverse KL projection only in a neighborhood of the sampling distribution. This rapid degradation under off-policy drift directly motivates the exact listwise projection proposed in LPO.

Remark: Connection to group-based policy gradients. Our off-policy analysis explicitly uncovers the structural relationship between exact listwise projection and practical group-based PG methods. By performing a strict first-order Taylor expansion on the exact reverse KL projection coefficient, we decouple it into three components:

$$c_k^{\mathrm{revKL}} \approx \underbrace{\frac{A_k}{K} + \frac{s_{\theta,k}A_k}{K}}_{\text{Pointwise advantage fitting}} \underbrace{- \left(\frac{\bar{s}_\theta A_k}{K} + \frac{1}{K^2} \sum_m s_{\theta,m} A_m \right)}_{\text{Listwise normalization}} \underbrace{- \frac{s_{\theta,k} - \bar{s}_\theta}{K}}_{\text{Intrinsic entropy regularization}} . \tag{29}$$

Remarkably, the gradient coefficient of group-based policy gradients without clipping, given by $c_k^{\mathrm{PG}} = \frac{r_k A_k}{K}$, yields a first-order expansion $c_k^{\mathrm{PG}} \approx \frac{A_k}{K} + \frac{s_{\theta,k}A_k}{K}$. This reveals a direct mathematical connection: the pointwise IS objective utilized in group-based policy gradients formally corresponds to the first-order advantage-fitting component of the reverse KL projection on the simplex, while the exact listwise formulation explicitly retains the coupled listwise normalization and intrinsic entropy regularization.

B.3 Proof of Theorem 1

Theorem 1 (Listwise Gibbs target)**.** *The objective $\hat{J}(w)$ in Eq. (7) has a unique maximizer w^*:*

$$w_k^* = \mathrm{softmax}(\phi)_k, \quad \textit{with } \phi_k = \frac{R_k}{\tau} + s_{t,k}. \tag{8}$$

Proof. Consider the optimization problem

$$w^* = \arg\max_{w \in \Delta^{K-1}} \hat{J}(w), \qquad \hat{J}(w) = \sum_{k=1}^K w_k R_k - \tau D_{\mathrm{KL}}(w \| P_t), \tag{30}$$

where $P_t = \mathrm{softmax}(s_t)$ satisfies $P_{t,k} > 0$ for all k.

Expanding the KL term gives

$$\hat{J}(w) = \sum_{k=1}^K w_k R_k - \tau \sum_{k=1}^K w_k \log \frac{w_k}{P_{t,k}}. \tag{31}$$

Introduce the Lagrangian for the simplex constraint $\sum_k w_k = 1$:

$$\mathcal{L}(w, \lambda) = \sum_{k=1}^K w_k R_k - \tau \sum_{k=1}^K w_k \log \frac{w_k}{P_{t,k}} + \lambda \left(1 - \sum_{k=1}^K w_k \right). \tag{32}$$

Setting the stationary condition $\partial \mathcal{L} / \partial w_k = 0$ yields

$$R_k - \tau(\log w_k - \log P_{t,k} + 1) - \lambda = 0, \tag{33}$$

hence

$$\log w_k = \frac{R_k}{\tau} + \log P_{t,k} - 1 - \frac{\lambda}{\tau}. \tag{34}$$

Exponentiating,

$$w_k = P_{t,k} \exp(R_k/\tau) \cdot C, \qquad C = \exp(-1 - \lambda/\tau). \tag{35}$$

Using $\sum_k w_k = 1$, the normalization constant is

$$C^{-1} = \sum_{j=1}^K P_{t,j} \exp(R_j/\tau), \tag{36}$$

therefore

$$w_k^* = \frac{P_{t,k} \exp(R_k/\tau)}{\sum_{j=1}^K P_{t,j} \exp(R_j/\tau)}. \tag{37}$$

Equivalently,

$$w^* = \mathrm{softmax}\big(R/\tau + \log P_t\big). \tag{38}$$

Since $\log P_t = s_t - \log \sum_j e^{s_{t,j}}$, softmax is shift-invariant, this yields

$$w^* = \mathrm{softmax}(R/\tau + s_t) = \mathrm{softmax}(\phi), \qquad \phi_k = \frac{R_k}{\tau} + s_{t,k}, \tag{39}$$

which is Eq. (8).

Finally, $\hat{J}(w)$ is strictly concave on Δ^{K-1}: the reward term is linear in w, while $D_{\mathrm{KL}}(w\|P_t)$ is strictly convex for $P_{t,k} > 0$. Hence the maximizer is unique. □

B.4 Proximal Objective as Reverse KL

Proposition 3 (Proximal objective as reverse KL)**.** $\hat{J}(P_\theta) = -\tau D_{\mathrm{KL}}(P_\theta\|w^*) + \tau \log \hat{Z}$, *so* $\arg\max_{P_\theta} \hat{J}(P_\theta) = \arg\min_{P_\theta} D_{\mathrm{KL}}(P_\theta\|w^*)$.

Proof. From Theorem 1, $w_k^* = P_{t,k} \exp(R_k/\tau)/\hat{Z}$ where $\hat{Z} = \sum_j P_{t,j} \exp(R_j/\tau)$. Therefore $\log w_k^* = R_k/\tau + \log P_{t,k} - \log \hat{Z}$. Expanding the reverse KL:

$$\begin{aligned}
-\tau D_{\mathrm{KL}}(P_\theta\|w^*) &= -\tau \sum_{k=1}^{K} P_{\theta,k} \log \frac{P_{\theta,k}}{w_k^*} \\
&= -\tau \sum_k P_{\theta,k} \left[\log P_{\theta,k} - \log w_k^*\right] \\
&= -\tau \sum_k P_{\theta,k} \left[\log P_{\theta,k} - R_k/\tau - \log P_{t,k} + \log \hat{Z}\right] \\
&= \sum_k P_{\theta,k} R_k - \tau \sum_k P_{\theta,k} \log \frac{P_{\theta,k}}{P_{t,k}} - \tau \log \hat{Z}.
\end{aligned}$$

Recognizing $\hat{J}(P_\theta) = \sum_k P_{\theta,k} R_k - \tau D_{\mathrm{KL}}(P_\theta\|P_t) = \sum_k P_{\theta,k} R_k - \tau \sum_k P_{\theta,k} \log(P_{\theta,k}/P_{t,k})$, we obtain $\hat{J}(P_\theta) = -\tau D_{\mathrm{KL}}(P_\theta\|w^*) + \tau \log \hat{Z}$. □

B.5 Proof of Theorem 2

Theorem 2 (Performance improvement bound)**.** *Assume* $|R_k| \le R_{\max}$. *If the projection step achieves* $\mathrm{TV}(P_{t+1}, w^*) \le \epsilon_{\mathrm{proj}}$, *then*

$$\hat{R}(P_{t+1}) \ge \hat{R}(P_t) + \underbrace{\tau\left[D_{\mathrm{KL}}(w^*\|P_t) + D_{\mathrm{KL}}(P_t\|w^*)\right]}_{\textit{target gain} \ge 0} - \underbrace{2R_{\max}\epsilon_{\mathrm{proj}}}_{\textit{projection error}}. \tag{9}$$

Proof. **(a)** By Proposition 3, $\hat{J}(w) = -\tau D_{\mathrm{KL}}(w\|w^*) + \tau \log \hat{Z}$. Evaluating at the anchor P_t:

$$\hat{R}(P_t) = \hat{J}(P_t) = \tau \log \hat{Z} - \tau D_{\mathrm{KL}}(P_t\|w^*). \tag{40}$$

Evaluating at the target w^* (where $D_{\mathrm{KL}}(w^*\|w^*) = 0$):

$$\hat{R}(w^*) = \hat{J}(w^*) + \tau D_{\mathrm{KL}}(w^*\|P_t) = \tau \log \hat{Z} + \tau D_{\mathrm{KL}}(w^*\|P_t). \tag{41}$$

Subtracting: $\hat{R}(w^*) - \hat{R}(P_t) = \tau[D_{\mathrm{KL}}(w^*\|P_t) + D_{\mathrm{KL}}(P_t\|w^*)]$.

(b) We bound the expected reward error using the Total Variation (TV) distance. By definition, the L_1 norm relates to the TV distance as $\|P_{t+1} - w^*\|_1 = 2\,\mathrm{TV}(P_{t+1}, w^*)$. By applying Pinsker's inequality, the TV distance is upper-bounded by either choice of KL projection:

$$\mathrm{TV}(P_{t+1}, w^*) \le \sqrt{\frac{1}{2} \min\big(D_{\mathrm{KL}}(w^*\|P_{t+1}),\ D_{\mathrm{KL}}(P_{t+1}\|w^*)\big)}. \tag{42}$$

Assuming the projection step achieves $\mathrm{TV}(P_{t+1}, w^*) \le \epsilon_{\mathrm{proj}}$, we apply Hölder's inequality with $|R_k| \le R_{\max}$:

$$|\hat{R}(P_{t+1}) - \hat{R}(w^*)| = \left|\sum_k (P_{t+1,k} - w_k^*) R_k\right| \le R_{\max} \|P_{t+1} - w^*\|_1 = 2R_{\max}\epsilon_{\mathrm{proj}}. \tag{43}$$

Substituting this error term back into the minorization inequality from part (a) yields the final bound.

(c) Combining (a) and (b):

$$\begin{aligned}\hat{R}(P_{t+1}) &\geq \hat{R}(w^*) - 2R_{\max}\epsilon_{\text{proj}} \\ &\geq \hat{R}(P_t) + \tau[D_{\text{KL}}(w^*\|P_t) + D_{\text{KL}}(P_t\|w^*)] - 2R_{\max}\epsilon_{\text{proj}}.\end{aligned}$$

□

B.6 Proof of Proposition 2

Proposition 2 (Idealized full-space convergence)**.** *Let $\pi_0(y) > 0$ for all y, and assume $R(y)$ is bounded. Under exact proximal updates $\pi_{t+1}(y) \propto \pi_t(y)\exp(R(y)/\tau)$, the iteration satisfies $\pi_t(y) \propto \pi_0(y)\exp(tR(y)/\tau)$ and $\mathbb{E}_{\pi_t}[R] \to \max_y R(y)$ as $t \to \infty$.*

Proof. By induction: the base case $t = 0$ is trivial. If $\pi_t(y) \propto \pi_0(y)\exp(tR(y)/\tau)$, then $\pi_{t+1}(y) \propto \pi_t(y)\exp(R(y)/\tau) \propto \pi_0(y)\exp((t+1)R(y)/\tau)$.

For convergence, consider any two responses y_1, y_2 with $R(y_1) > R(y_2)$:

$$\frac{\pi_t(y_1)}{\pi_t(y_2)} = \frac{\pi_0(y_1)}{\pi_0(y_2)}\,\exp\big(t\cdot\frac{R(y_1)-R(y_2)}{\tau}\big) \to \infty. \tag{44}$$

Since $\pi_0(y) > 0$ for all y, the mass concentrates on $\arg\max_y R(y)$, giving $\mathbb{E}_{\pi_t}[R] \to \max_y R(y)$. □

Connecting global optimality to LPO. Proposition 2 characterizes the ideal full-space proximal operator: if one could exactly apply the Gibbs update over the entire response space, the resulting iteration converges to the global RL optimum. For autoregressive LLMs, however, the required partition function is intractable over the combinatorially large sequence space. This computational barrier motivates LPO. Rather than operating in the full space, LPO restricts the same target-projection principle to the finite response simplex induced by K sampled trajectories, yielding a principled and fully tractable approximation to the ideal proximal step.

B.7 Proof of Corollary 1

Corollary 1 (Gradient coefficient properties)**.** *The forward KL gradient coefficients c_k^{fwd} satisfy: (a) bounded: $|c_k^{\text{fwd}}| \leq 1$; (b) zero-sum: $\sum_k c_k^{\text{fwd}} = 0$; (c) self-correcting: $c_k^{\text{fwd}} \to 0$ as $P_\theta \to w^*$.*

Proof. Let $c_k^{\text{fwd}} = P_{\theta,k} - w_k^*$ where $P_\theta, w^* \in \Delta^{K-1}$.

(a) Since $P_{\theta,k} \in [0,1]$ and $w_k^* \in [0,1]$, we have $c_k^{\text{fwd}} \in [-1,1]$, hence $|c_k^{\text{fwd}}| \leq 1$.

(b) Since both P_θ and w^* are probability distributions, $\sum_{k=1}^K c_k^{\text{fwd}} = \sum_k P_{\theta,k} - \sum_k w_k^* = 1 - 1 = 0$. Partitioning into positive and negative parts: $\sum_{c_k^{\text{fwd}}>0} c_k^{\text{fwd}} = -\sum_{c_k^{\text{fwd}}<0} c_k^{\text{fwd}}$. Therefore $\sum_k |c_k^{\text{fwd}}| = 2\sum_{c_k^{\text{fwd}}>0} c_k^{\text{fwd}}$. Since each $c_k^{\text{fwd}} \leq 1$ and the positive parts sum to at most 1 (because $\sum_{c_k^{\text{fwd}}>0} c_k^{\text{fwd}} \leq \sum_{c_k^{\text{fwd}}>0} P_{\theta,k} \leq 1$), we obtain $\sum_k |c_k^{\text{fwd}}| \leq 2$.

(c) As $P_\theta \to w^*$, $c_k^{\text{fwd}} = P_{\theta,k} - w_k^* \to 0$ for all k by definition.

For the parameter-space bound, $\nabla_\theta \mathcal{L}_{\text{LPO}_{\text{fwd}}} = \sum_k c_k^{\text{fwd}} \nabla_\theta \log \pi_\theta(y_k|x)$, so by the triangle inequality: $\|\nabla_\theta \mathcal{L}_{\text{LPO}_{\text{fwd}}}\| \leq \sum_k |c_k^{\text{fwd}}| \cdot \|\nabla_\theta \log \pi_\theta(y_k|x)\| \leq 2G_{\max}$. □

B.8 Proof of Corollary 2

Corollary 2 (Mode-Coverage)**.** *If $w_k^* \geq \alpha$ and $D_{\text{KL}}(w^*\|P_\theta) \leq D$, then $P_{\theta,k} > \alpha\exp(-D/\alpha - 1)$.*

Proof. To rigorously bound $P_{\theta,k}$, we construct a binary event space (whether a response is k or not k). By the Data Processing Inequality (DPI), the binary KL divergence is bounded by the full KL

divergence:

$$w_k^* \log \frac{w_k^*}{P_{\theta,k}} + (1 - w_k^*) \log \frac{1 - w_k^*}{1 - P_{\theta,k}} \leq D_{\mathrm{KL}}(w^* \| P_\theta) \leq D. \tag{45}$$

Since $1 - P_{\theta,k} \leq 1$, the term $\log(1/(1 - P_{\theta,k})) \geq 0$. Dropping this non-negative component preserves the upper bound inequality:

$$w_k^* \log \frac{w_k^*}{P_{\theta,k}} + (1 - w_k^*) \log(1 - w_k^*) \leq D. \tag{46}$$

Rearranging the terms to isolate $P_{\theta,k}$, we obtain:

$$\log \frac{w_k^*}{P_{\theta,k}} \leq \frac{D}{w_k^*} - \frac{1 - w_k^*}{w_k^*} \log(1 - w_k^*). \tag{47}$$

Exponentiating both sides yields the rigorously corrected lower bound:

$$P_{\theta,k} \geq w_k^* \exp\left(-\frac{D}{w_k^*}\right) (1 - w_k^*)^{\frac{1 - w_k^*}{w_k^*}}. \tag{48}$$

Let $f(x) = x \exp(-D/x)(1 - x)^{\frac{1-x}{x}}$. It can be shown that $f(x)$ is monotonically increasing for $x \in (0, 1)$. Given the assumption that $w_k^* \geq \alpha > 0$, it follows that $P_{\theta,k} \geq f(w_k^*) \geq f(\alpha)$. Therefore, we conclude:

$$P_{\theta,k} \geq \alpha \exp\left(-\frac{D}{\alpha}\right) (1 - \alpha)^{\frac{1-\alpha}{\alpha}} \geq \alpha \exp\left(-\frac{D}{\alpha} - 1\right). \tag{49}$$

□

C Additional Discussions

C.1 Contribution Clarification

This work mainly makes two contributions: (i) the Target-Projection (TP) framework, a unified geometric interpretation showing that dominant group-based PG methods implicitly construct the same Gibbs target family and approximate a reverse KL projection toward it; and (ii) Listwise Policy Optimization (LPO), which makes this target-projection explicit and, by decoupling the target from the projection, opens divergence selection as a new design axis inaccessible under the implicit PG paradigm, with provable theoretical guarantees and consistent empirical gains.

Several clarifications are included regarding the scope and design choices.

1. The TP analysis and LPO operate in the group-based regime ($K \geq 2$), which covers the vast majority of contemporary RLVR practice. Single-sample methods (K=1) lack a per-prompt simplex and require a different analytical treatment, as discussed in Appendix C.2.
2. The specific choice of forward or reverse KL is not a core contribution of this work, with the broader design space discussed in Appendix C.6. Reverse KL is a natural choice since policy gradient implicitly performs an approximate reverse KL projection (Proposition 1). Forward KL is similarly motivated by its mode-covering geometry, whose benefit for diversity has been observed in adjacent settings [Wang et al., 2023, Li et al., 2025, Anthony et al.].
3. Recent engineering innovations, e.g., dynamic sampling [Yu et al., 2025] or predictive sampling [Wang et al., 2026, Qu et al., 2025c,b], are orthogonal to LPO and can be combined with it, as discussed in Appendix C.3. The current experiments intentionally use a minimal shared pipeline to cleanly attribute gains to the target-projection mechanism.
4. The experiments adopt a paired-temperature design, varing only the projection mechanism to isolate it as the sole controlled variable. While the temperature τ could be independently tuned, this is deliberately avoided to ensure a fair comparison, leaving for future work.
5. All theoretical results and the implementation hold for arbitrary rewards $R_k \in \mathbb{R}$ without binary assumption. The focus on binary outcome rewards reflects the dominant RLVR setting, while the programming experiments already assess a non-binary reward.

6. The experimental evaluation focuses on reasoning tasks with verifiable rewards, the primary application domain of group-based PG methods. The TP analysis and LPO are not inherently limited to this setting: extending empirical validation to broader RL post-training scenarios, e.g., RLHF with learned reward models, is a natural direction for future work.

C.2 Extensions and Future Directions

Step-level listwise projection. Real-world applications often necessitate fine-grained optimization beyond sequence-level rewards, such as multi-turn agentic RL [Jin et al., 2025, Qu et al., 2025a, 2024, Zou et al., 2026a] or reasoning tasks with dense step-level rewards [Lightman et al., 2023]. The current sequence-level framework may extend to these scenarios: given a shared intermediate state, one can sample K candidate continuations to form the local response simplex. Crucially, deriving the target for these immediate steps requires estimating their expected final outcomes. This can be achieved by rolling out each continuation to the terminal state. Alternatively, to bypass the prohibitive cost of full rollouts, one can rely on a value network or a value-calibrated process reward model [Lightman et al., 2023] to estimate these expected future returns. In either setting, the core LPO machinery carries over naturally, shifting the primary practical challenge to the fidelity of the step-level value estimation.

Off-policy replay. Because the listwise projection operates on the response simplex for each prompt, LPO can theoretically incorporate off-policy data [Mao et al., 2026b] to improve sample efficiency and amortize the high rollout costs typical of RLVR. Specifically, by recording the behavior policy π_b used to generate past responses, LPO can account for off-policy drift via importance sampling ratios π_t/π_b and π_θ/π_b. The listwise normalization implicitly acts as a self-normalized importance sampling (SNIS) estimator, inherently adapting the policy and target distribution without altering the underlying projection geometry. Despite this theoretical elegance, realizing off-policy replay introduces practical optimization hurdles. As the policy evolves, severe drift from stale checkpoints can yield extreme probability ratios, which may collapse the effective listwise distributions and destabilize the projection gradients. Developing robust staleness-filtering or trust-region buffer management strategies to stabilize off-policy LPO remains a promising direction for future work.

Beyond group-based sampling. Current LPO requires $K \geq 2$ responses per prompt to form the response simplex, which precludes direct application in single-sample ($K{=}1$) pipelines. One potential resolution is to assemble virtual groups using the off-policy replay buffer, though this inherits the aforementioned stability challenges. A minimal alternative constructs a virtual response simplex by pairing the single sampled reward R with a batch-level baseline b. This contrastive formulation yields a sigmoid-squashed gradient coefficient $c = \frac{1}{2} - \sigma\big((R-b)/\tau\big)$ that preserves boundedness ($|c| \leq 1/2$), though it necessarily sacrifices the zero-sum property as there is no physical group. Both relaxations remain exploratory and characterizing their practical tradeoffs is a promising avenue.

Alternative divergences and adaptive scheduling. A distinctive feature of the explicit target-projection framework is the complete decoupling of the target distribution from the projection divergence, which is a critical design axis unavailable to policy gradient methods. This separation naturally invites the exploration of entirely different statistical divergences that might induce unique and favorable optimization geometries tailored to specific reasoning tasks, as analyzed in Appendix C.6. Furthermore, this decoupling enables dynamic scheduling strategies during training. For instance, one could employ forward KL in early stages to encourage broad exploration, and subsequently switch to reverse KL for stable late-stage exploitation, or progressively anneal the temperature τ to sharpen the target as the performance improves. Systematic exploration of this expanded design space constitutes a natural next step for RL post-training.

C.3 Existing Group-based RLVR as Implicit Target-Projection

As revealed in Section 3, the dominant group-based RL algorithms can be unified under a shared geometric structure: each defines an implicit Gibbs target distribution w^* and executes an approximate reverse KL projection via policy gradient. The methods differ primarily in how they normalize advantages, which determines the implicit temperature τ and thus the sharpness of w^*. Table 2 groups these methods by their implicit target family.

Table 2: Target-Projection decomposition of existing methods, grouped by implicit target family.

Target family	Methods	τ
$\mathrm{softmax}(R/\sigma_G)$	GRPO, DAPO, CISPO, GSPO	σ_G
$\mathrm{softmax}(R)$	Dr.GRPO, RLOO ($\tau \approx 1$), ReMax	1
$\mathrm{softmax}(R/\mu_G)$	MaxRL	μ_G
$\mathrm{softmax}(R/\sigma_{B'})$	REINFORCE++$_{w/Baseline}$	$\sigma_{B'}$

σ_G-family: GRPO [Shao et al., 2024b], DAPO [Yu et al., 2025], CISPO [Chen et al., 2025], GSPO [Zheng et al., 2025].

$$A_k = \frac{R_k - \mu_G}{\sigma_G}, \qquad \tau = \sigma_G = \sqrt{\mu_G(1-\mu_G)}, \qquad w^* = \mathrm{softmax}(R/\sigma_G). \tag{50}$$

The temperature is adaptive: maximal at $\mu_G = 0.5$ (balanced groups) and vanishing as $\mu_G \to 0$ or 1 (near-unanimous groups), coupling target sharpness to group difficulty. DAPO adds four projection-level innovations: asymmetric clipping, dynamic sampling to filter uninformative groups, token-level loss normalization, and overlong reward shaping. CISPO modifies the projection by replacing clipping with a stop-gradient on the clipped importance ratio, preserving gradient contributions from all tokens. GSPO lifts the importance ratio and clipping from the token level to the sequence level $s_k = [\pi_\theta(y_k|x)/\pi_{\theta_{\text{old}}}(y_k|x)]^{1/|y_k|}$, aligning the optimization unit with the reward granularity. Many of these projection-level engineering tricks are orthogonal to our target construction and can be seamlessly integrated into the LPO framework.

$\tau \approx 1$ family: Dr.GRPO [Liu et al., 2025b], RLOO [Ahmadian et al., 2024], ReMax [Li et al., 2023].

$$A_k = R_k - \mu, \qquad \tau = 1, \qquad w^* = \mathrm{softmax}(R). \tag{51}$$

Dr.GRPO removes σ_G normalization (fixing $\tau = 1$) and adopts token-level loss normalization to address length bias. RLOO uses a leave-one-out baseline ($\tau = (K-1)/K \to 1$), yielding an unbiased advantage estimator with nearly the same implicit target. ReMax uses a greedy-decode baseline R_{greedy} which cancels in the softmax, recovering the same target.

MaxRL [Tajwar et al., 2026]. $A_k = (R_k - \mu_G)/\mu_G$, $\tau = \mu_G = n/K$, $w^* = \mathrm{softmax}(R/\mu_G)$. The temperature is directly proportional to the success rate, implementing an implicit curriculum: hard prompts (low μ_G) receive aggressively sharp targets to encourage exploitation, while easy prompts (high μ_G) receive diffuse targets to maintain diversity.

REINFORCE++ [Hu, 2025]. REINFORCE++ proposes two variants. The base variant uses single-stage batch normalization $A_k = (R_k - \mu_B)/\sigma_B$; its primary use case is $K{=}1$, where no per-prompt group exists and the target-projection decomposition does not apply. The *w/ Baseline* variant employs a two-stage process: first subtract the per-group mean to reshape rewards, $A'_k = R_k - \mu_G$, then normalize by the global batch statistics of these centered rewards, $A_k^{\text{norm}} = (A'_k - \mu_{B'})/\sigma_{B'}$. Since both μ_G and $\mu_{B'}$ are constant within a group, they cancel under softmax, yielding $w^* = \mathrm{softmax}(R/\sigma_{B'})$ with $\tau = \sigma_{B'}$. Here $\sigma_{B'}$ is the batch-level standard deviation of the group-centered rewards.

C.4 Listwise vs. Pointwise Projection

An alternative to the listwise framework developed in Section 4 is standard *pointwise* projection, a paradigm widely used in classical RL algorithms (e.g., MPO [Abdolmaleki et al., 2018] and AWR [Peng et al., 2019]). Both paradigms share the same target step, constructing the reward-weighted Gibbs distribution $w_k^* \propto \pi_b(y_k)\exp(R_k/\tau)$, but they diverge fundamentally in how they project the policy toward it.

Independent vs. coupled formulation. Pointwise projection minimizes a weighted negative log-likelihood:

$$\mathcal{L}_{\text{pointwise}} = -\sum_{k=1}^{K} w_k^* \log \pi_\theta(y_k|x), \tag{52}$$

which treats each sampled response independently. The gradient coefficient for response k is simply $c_k^{\text{point}} = -w_k^*$. This yields a strictly one-sided update that pushes probability mass *toward* high-weight responses without any coupled counterbalancing force.

In contrast, LPO with forward KL minimizes divergence $D_{\text{KL}}(w^* \| P_\theta)$, where $P_\theta = \text{softmax}(s_\theta)$ is the policy's listwise distribution. This explicit listwise formulation couples all K responses through a shared normalization factor. Consequently, the gradient coefficient $c_k = P_{\theta,k} - w_k^*$ is strictly two-sided: responses where the policy over-allocates probability mass ($P_{\theta,k} > w_k^*$) are actively suppressed, while under-allocated responses are boosted.

Structural consequences. This architectural difference in the projection space produces three structural properties that pointwise projection inherently lacks:

1. **Zero-sum updates.** For LPO, the coefficients strictly sum to zero: $\sum_k c_k = 0$, acting as a built-in control variate for variance reduction. For pointwise projection, $\sum_k c_k^{\text{point}} = -\sum_k w_k^* = -1$, yielding a net gradient direction that exerts a continuous, uncalibrated pull on the parameter space.
2. **Bounded gradients.** LPO coefficients satisfy $\sum_k |c_k| \leq 2$ (Corollary 1), providing an intrinsic, reward-scale-invariant bound on the projection step. Pointwise projection lacks this relative scaling.
3. **Self-correcting convergence.** As $P_\theta \to w^*$, the LPO coefficients vanish ($c_k = P_{\theta,k} - w_k^* \to 0$), meaning optimization naturally terminates once the target is matched. Pointwise coefficients ($c_k^{\text{point}} = -w_k^*$) are constant with respect to π_θ.

Origin of the difference. The pointwise objective $-\sum_k w_k^* \log \pi_\theta(y_k)$ mathematically corresponds to the cross-entropy $H(w^*, \pi_\theta)$, which equals $D_{\text{KL}}(w^* \| \pi_\theta) + H(w^*)$. Because π_θ is evaluated independently per response and is not normalized over the response group, this KL divergence measures the gap between unnormalized densities. LPO, by contrast, operates on the normalized listwise distribution $P_\theta \in \Delta^{K-1}$, which lives on the exact same finite probability simplex as w^*. This shared simplex geometry is what dictates the two-sided, zero-sum gradient structure.

Connection to Expectation-Maximization. The explicit target-projection procedure mirrors the structure of the Expectation-Maximization (EM) algorithm [Dayan and Hinton, 1997, Neal and Hinton, 1998]: the Gibbs target construction resembles an E-step that forms a target distribution, while the divergence minimization corresponds to an M-step that fits the model to this target.

C.5 Connection to DPO and Preference Optimization

When $K = 2$, LPO reduces to a pairwise objective closely related to Direct Preference Optimization (DPO) [Rafailov et al., 2024]. Consider two responses: a preferred response y_w with reward $R_w = 1$, and a dispreferred response y_l with reward $R_l = 0$.

For two responses, the listwise distribution becomes

$$P_w = \frac{\exp(s_w)}{\exp(s_w) + \exp(s_l)} = \sigma(s_w - s_l), \tag{53}$$

where $s_k = \log(\pi_\theta(y_k|x)/\pi_b(y_k|x))$ and $\sigma(\cdot)$ is the sigmoid function. In on-policy setup ($\pi_t = \pi_b$), the baseline distribution is uniform, yielding

$$w_w^* = \sigma(1/\tau), \qquad w_l^* = \sigma(-1/\tau). \tag{54}$$

The forward-KL objective then simplifies to

$$\mathcal{L}_{\text{LPO}_{\text{fwd}}} = -\sigma(1/\tau) \log \sigma(s_w - s_l) - \sigma(-1/\tau) \log \sigma(s_l - s_w), \tag{55}$$

which is a binary cross-entropy objective with temperature-controlled soft targets.

By comparison, DPO uses the pairwise logistic objective

$$\mathcal{L}_{\text{DPO}} = -\log \sigma\big(\beta(s_w - s_l)\big), \tag{56}$$

where $s_k = \log(\pi_\theta(y_k|x)/\pi_{\text{ref}}(y_k|x))$.

Thus, both methods share the same pairwise sigmoid structure, but differ fundamentally in four aspects: (i) standard DPO operates within an offline paradigm on static datasets, whereas LPO is an online RL algorithm; (ii) DPO penalizes logits against a static reference policy π_{ref}, whereas LPO derives its target within a trust region around the pre-update policy π_t; (iii) DPO is derived under a Bradley–Terry style preference model, whereas LPO arises from explicit divergence projection on the response simplex; (iv) LPO uses soft targets controlled by τ, recovering a hard preference target as $\tau \to 0$.

This view places DPO-style pairwise optimization as the foundational $K = 2$ baseline of the broader LPO framework, which naturally extends from pairwise preferences ($K = 2$) to listwise optimization ($K > 2$), and further to the population-level RL-as-inference limit as $K \to \infty$.

Distinction from Listwise Preference Optimization. Recent works like LiPO [Liu et al., 2025a] extend DPO to listwise preference optimization with Plackett–Luce style ranking models. Despite the similar "listwise" terminology, these methods learn from ranked preference data $y_1 \succ y_2 \succ \cdots \succ y_K$. In contrast, LPO is designed for online RLVR, utilizing absolute reward signals for explicit target-projection without any ranking-model assumptions. Mathematically, Plackett–Luce ranking models and our Gibbs target both take a normalized softmax form. Thus, they represent complementary ways of obtaining the same exponential-family target: one inferred from comparisons, the other directly specified by rewards.

C.6 Extension to General Divergences

In Section 4.2, we instantiated LPO using forward and reverse KL divergences. However, the projection framework is not specific to KL and can be applied to any differentiable divergence defined on the probability simplex, including general f-divergences such as the Jensen–Shannon divergence.

Let $\mathcal{L} = D(w^*, P_\theta)$ be a differentiable divergence on Δ^{K-1}. The gradient takes the form

$$\nabla_\theta \mathcal{L} = \sum_{k=1}^{K} c_k \nabla_\theta \log \pi_\theta(y_k \mid x), \tag{57}$$

where the gradient coefficient is $c_k = \partial \mathcal{L} / \partial s_{\theta,k}$. Applying the chain rule with the softmax Jacobian, we have

$$c_k = \sum_{j=1}^{K} \frac{\partial \mathcal{L}}{\partial P_{\theta,j}} \frac{\partial P_{\theta,j}}{\partial s_{\theta,k}} = P_{\theta,k} \frac{\partial \mathcal{L}}{\partial P_{\theta,k}} - P_{\theta,k} \sum_{j=1}^{K} P_{\theta,j} \frac{\partial \mathcal{L}}{\partial P_{\theta,j}}. \tag{58}$$

Summing these coefficients over all K responses yields the identity

$$\sum_{k=1}^{K} c_k = \sum_{k=1}^{K} P_{\theta,k} \frac{\partial \mathcal{L}}{\partial P_{\theta,k}} - \underbrace{\left(\sum_{k=1}^{K} P_{\theta,k} \right)}_{=1} \left(\sum_{j=1}^{K} P_{\theta,j} \frac{\partial \mathcal{L}}{\partial P_{\theta,j}} \right) = 0. \tag{59}$$

This zero-sum property is a direct consequence of the softmax parameterization on the probability simplex and holds for any differentiable objective defined on P_θ. It plays a role analogous to a baseline in policy gradient methods and contributes to stabilizing the update.

While this zero-sum property is universal, other characteristics, such as coefficient boundedness or mode-seeking behavior, depend on the specific choice of divergence. KL divergences are adopted as natural default choices in LPO due to their stability and well-understood geometry.

C.7 Entropy Regularization and Reverse KL Diversity

Reverse KL as max-entropy RL. The objective of LPO with reverse KL is equivalent to $\max_\theta \sum_k P_{\theta,k} \phi_k + H(P_\theta)$, mirroring the maximum entropy RL objective [Ziebart, 2010]. Here, the explicit entropy bonus emerges naturally from the structural formulation of the divergence. In contrast, standard policy gradient methods lose this property, as they are equivalent to a first-order approximation only at the on-policy point.

Entropy regularization as target mixing. Adding entropy bonus $\gamma H(\pi_\theta)$ modifies the listwise target to $\tilde{w}^* = \mathrm{softmax}(R/(\tau+\gamma))$ in the on-policy setup, equivalent to increasing τ by γ. The entropy bonus is redundant when τ is a controllable hyperparameter.

C.8 Broader Societal Impacts

This work introduces LPO as a novel paradigm for RLVR. As an algorithmic contribution to policy optimization, LPO may improve the efficiency and stability of RL post-training, potentially reducing the computational cost of training strong LLMs. More broadly, improvements in reasoning capability and training efficiency may indirectly benefit downstream applications of LLMs, such as scientific problem solving, software development, and educational tools, by enabling more capable and reliable systems. On the negative side, the method inherits the general risks associated with increasingly capable LLMs, including potential dual-use concerns if deployed without appropriate safeguards. Additionally, while LPO improves optimization efficiency in RLVR, addressing the environmental and societal costs of large-scale model training remains an open challenge for the broader research community.

D Implementation Details

D.1 Tasks

D.1.1 Logical Reasoning

Training Dataset. We adopt the Countdown Number Game as the logical reasoning testbed. This task requires models to synthesize basic arithmetic operations to reach a target value using a provided set of integers. We use a subset of 2000 problems sampled from the Countdown 34 dataset (https://huggingface.co/datasets/Jiayi-Pan/Countdown-Tasks-3to4) as training dataset, which supplies either three or four source numbers per question.

Evaluation Benchmarks. We assess model performance using two reserved evaluation sets: a split of 512 instances from Countdown 34 (CD34) and a subset of 512 instances from Countdown 4 (CD4), available at https://huggingface.co/datasets/Jiayi-Pan/Countdown-Tasks-4. The CD4 variant is notably more difficult because it strictly guarantees four source numbers per problem, thereby massively expanding the combinatorial search space. To evaluate performance, we generate 64 independent responses per instance to compute both the expected `Pass@1` (the average correctness across all 64 samples) and the `Pass@64` metrics. All reported training curves reflect the average performance across both the CD34 and CD4 benchmarks.

Reward Function. Following Pan et al. [2025], we augment the binary accuracy reward with a formatting bonus. This design explicitly incentivizes proper structural adherence alongside correct reasoning:

$$r = \begin{cases} 1 & \text{if the response is correct,} \\ 0.1 & \text{if the response is incorrect but properly formatted,} \\ 0 & \text{otherwise.} \end{cases} \tag{60}$$

D.1.2 Mathematics Reasoning

Training Dataset. Following Qu et al. [2025b], Mao et al. [2026a], we train models on the MATH dataset [Hendrycks et al., 2021], which consists of 7.5k problems from mathematics competitions. We use the public version hosted at https://huggingface.co/datasets/DigitalLearningGmbH/MATH-lighteval. For extended validation at a larger scale, we additionally train the Qwen3-14B-Base model on the Polaris [An et al., 2025] dataset in Appendix E.1, which comprises a broader collection of roughly 53k high-quality mathematical reasoning problems. The Polaris dataset is hosted at https://huggingface.co/datasets/POLARIS-Project/Polaris-Dataset-53K.

Evaluation Benchmarks. Following Gao et al. [2025], Qu et al. [2026], we evaluate mathematical reasoning performance on a suite of benchmarks, including AIME24, AIME25, AMC23, MATH500 [Lightman et al., 2023], Minerva Math [Lewkowycz et al., 2022], and OlympiadBench [He

et al., 2024], using the datasets hosted at https://huggingface.co/datasets/math-ai. Following prior works [Gao et al., 2025, Qu et al., 2026], we sample k independent responses per problem to compute both the expected Pass@1 (defined as the average accuracy across all k samples, or avg@k) and the Pass@k metrics. The sample size k is tailored to the size and difficulty of each benchmark: we set $k = 32$ for competition-level suites (AIME24, AIME25, AMC23), $k = 4$ for Minerva Math, and $k = 1$ for MATH500 and OlympiadBench. Training curves report the average performance across all math benchmarks.

Reward Function. Following the default configuration in verl [Sheng et al., 2024], we use a binary reward function that assigns 1 to correct responses and 0 otherwise.

D.1.3 Programming

Training Dataset. To assess the training performance in code generation, following Cui et al. [2025], we adopt the code split in PRIME dataset, available at https://huggingface.co/datasets/PRIME-RL/Eurus-2-RL-Data, which contains 25.3k problems that are mainly programming competition level.

Evaluation Benchmarks. For evaluation, we evaluate on the 1k held-out validation problems from the PRIME code dataset. For each prompt, we sample $k = 8$ independent Python programs to compute the expected Pass@1 (the average success rate across the 8 samples) and the pass@8 metrics.

Reward Function. Following PRIME [Cui et al., 2025], we extract the generated Python program and evaluate it against a suite of test cases. The reward is defined as the fraction of tests passed:

$$r = \frac{\text{number of passed tests}}{\text{total number of tests}}. \tag{61}$$

Compared to a strict binary reward setup, this formulation provides a denser learning signal, yielding values in $[0, 1]$ where 1 indicates a fully correct solution and 0 indicates complete failure.

D.1.4 Geometry

Training Dataset. We train on the 2.1k-problem training split of the Geometry3k dataset [Lu et al., 2021, Hiyouga, 2025], available at https://huggingface.co/datasets/hiyouga/geometry3k. Each problem in Geometry3k consists of a geometric diagram and an accompanying natural language question, often requiring multi-step spatial or logical reasoning.

Evaluation Benchmarks. We evaluate performance on the official 601-problem test split of Geometry3k. For each prompt, we generate 16 independent responses to calculate both the expected Pass@1 (the average accuracy across 16 samples) and the Pass@16 metrics.

Reward Function. Following verl [Sheng et al., 2024], we use the same reward function as in Countdown.

Appendix F presents data examples from each of the training datasets.

D.2 Models

We evaluate eight models spanning a diverse range of types, parameter scales, and model families. All models are sourced directly from their official Hugging Face repositories and used as released:

- Qwen3-1.7B-Base: https://huggingface.co/Qwen/Qwen3-1.7B-Base;
- Qwen3-4B-Base: https://huggingface.co/Qwen/Qwen3-4B-Base;
- Qwen3-8B-Base: https://huggingface.co/Qwen/Qwen3-8B-Base;
- Qwen3-14B-Base: https://huggingface.co/Qwen/Qwen3-14B-Base;
- Qwen2.5-VL-3B-Instruct: https://huggingface.co/Qwen/Qwen2.5-VL-3B-Instruct;

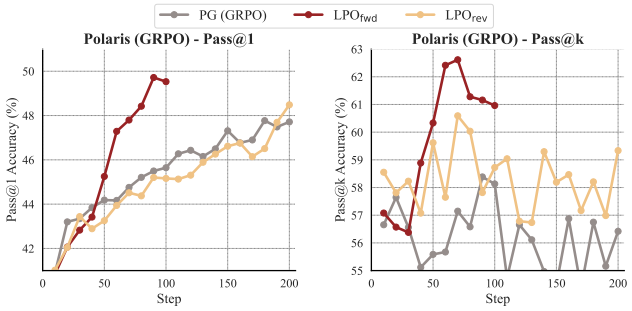


Figure 8: Scalability validation. We compare LPO with GRPO by training Qwen3-14B-Base on the larger Polaris dataset.

- DeepSeek-R1-Distill-Qwen-1.5B: https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-1.5B;
- Llama-3.1-8B-Instruct: https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct;
- Mistral-7B-Instruct-v0.1: https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.1;

D.3 Training Details

We evaluate our method against three representative group-based policy gradient baselines: GRPO [Shao et al., 2024b], Dr.GRPO [Liu et al., 2025b], and MaxRL [Tajwar et al., 2026], all implemented within the verl framework [Sheng et al., 2024]. Across all four reasoning scenarios, we sample a group of $K = 8$ responses per prompt during training to estimate advantages or construct response simplex. The generation temperature is set to 1.0 with `top_p` $= 1.0$, `top_k` $= -1.0$, and we disable the KL penalty by setting $\beta = 0$, consistent with Yu et al. [2025]. Evaluation generations use a lower temperature of 0.6, following common practice [Qu et al., 2025b].

We tailor the batch sizes and context lengths according to the complexity of the specific benchmark. For the Math and PRIME-Code tasks, we set the training batch size to 256, the mini-batch size to 128, and the maximum response length to 4096 tokens. For the Countdown and Geometry tasks, we scale down the training batch size to 128 and the mini-batch size to 64, with the maximum response length capped at 1024 tokens. This configuration performs two gradient updates per iteration, inherently introducing a mild off-policy drift. A strictly fully on-policy ablation is provided in Appendix E.4.

Optimization is uniformly performed using Adam [Kingma and Ba, 2014] with a learning rate of $1\mathrm{e}{-6}$ across all tasks. The optimizer parameters are set to $\beta = (0.9, 0.999)$ with a weight decay of 0.1. The clipping parameter is fixed at $\epsilon = 0.2$. Given the highly non-linear parameter-space updates, we additionally apply token-level clipping [Schulman et al., 2017b]. The token-level log-density ratio $\delta_{k,i} = \log \pi_\theta(y_{k,i}|x, y_{k,<i}) - \log \pi_b(y_{k,i}|x, y_{k,<i})$ is clipped to $[\log(1-\epsilon), \log(1+\epsilon)]$ and then weighted by c_k to form the final loss.

All experiments are conducted on 8 NVIDIA H20 GPUs.

E Extended Experimental Results

E.1 Scalability Validation

To verify the scalability and extensibility of the LPO framework, we conduct additional experiments using the Qwen3-14B-Base model on the Polaris dataset, which contains approximately 53k complex reasoning problems. We compare both LPO variants with the GRPO baseline. As shown in Fig. 8, LPO-fwd exhibits remarkable sample efficiency, reaching the peak performance achieved by GRPO at 200 training steps within only 70 steps, while simultaneously providing significant improvements in both Pass@1 and Pass@k metrics. For the LPO-rev variant, although its Pass@1 accuracy is comparable to GRPO, it shows superior robustness in maintaining Pass@k, effectively preserving response diversity. These findings provide strong evidence that LPO is scalable and capable of maintaining its theoretical advantages alongside increases in model capacity and data volume.

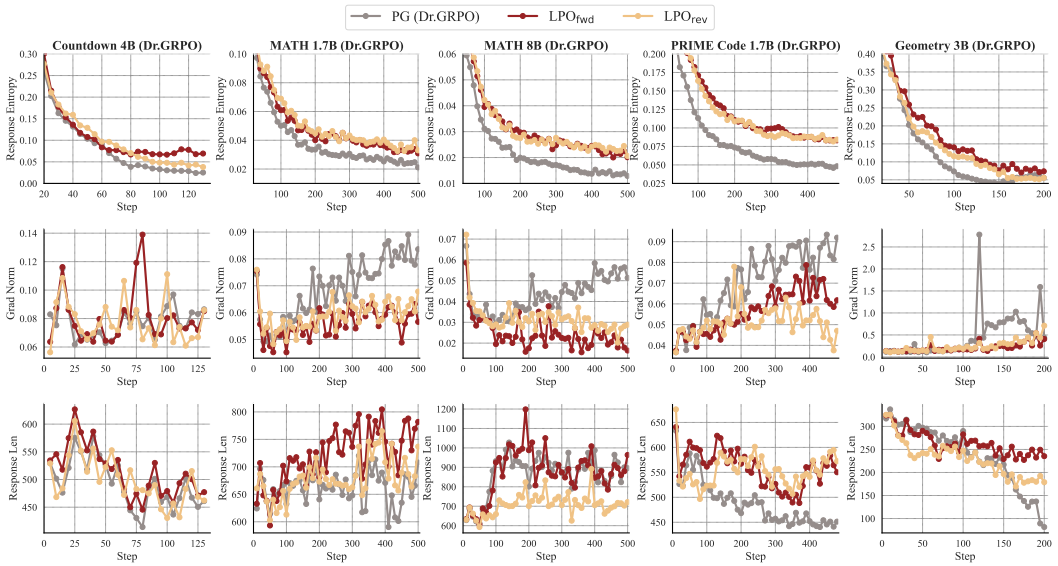


Figure 9: Training dynamics of LPO variants and Dr.GRPO. Rows from top to bottom respectively show the curves of response entropy, gradient norms, and response lengths.

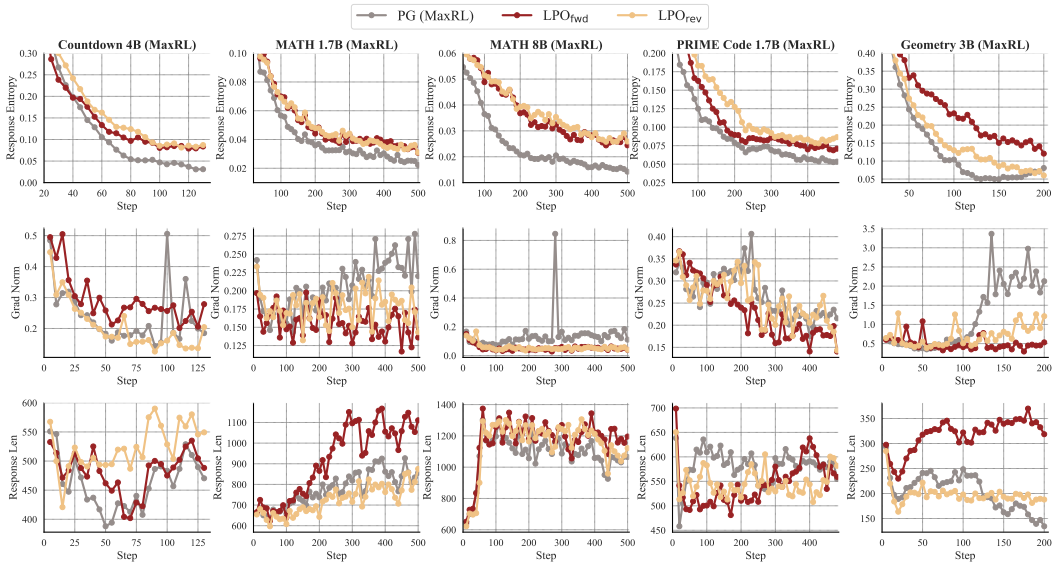


Figure 10: Training dynamics of LPO variants and MaxRL. Rows from top to bottom respectively show the curves of response entropy, gradient norms, and response lengths.

E.2 Extended Training Dynamics

To corroborate the analysis presented in Sec. 5.3, we provide the extended training dynamics of LPO compared against Dr.GRPO in Fig. 9 and MaxRL in Fig. 10.

Consistent with the observations relative to the GRPO baseline in the main text, LPO variants demonstrate superior optimization properties across these baselines. Specifically, LPO maintains higher response entropy, exhibits lower and more stable gradient norms, and encourages longer response chains. These supplementary results further support the structural advantages of listwise projection.

E.3 Generalization across LLM Families

To evaluate the generalizability of LPO, we conduct experiments across four prominent LLM families: Qwen, DeepSeek, Mistral, and Llama. These include models with different training paradigms such as base (only pre-trained), distilled, and instruction-tuned variants. As shown in Fig. 11, across

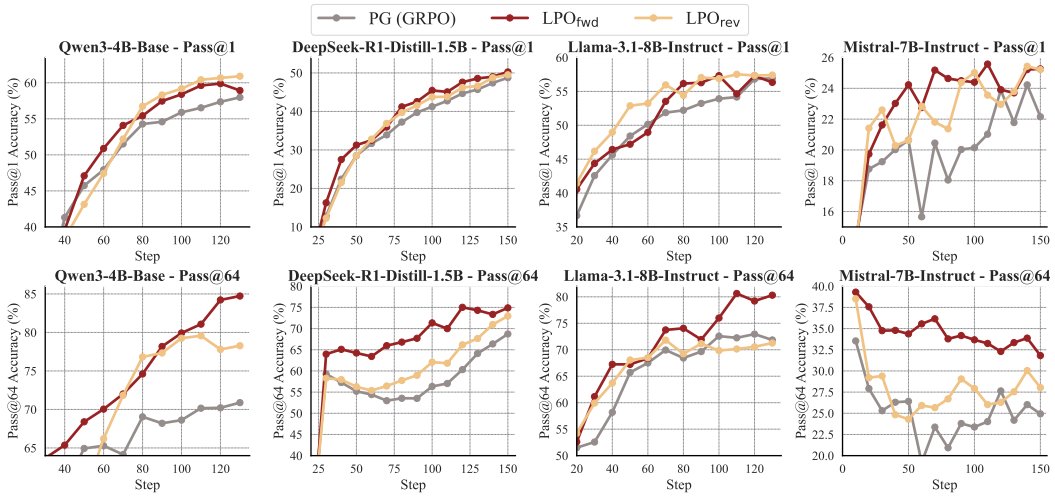


Figure 11: Generalization of LPO across diverse LLM families. Performance is evaluated on Countdown using Qwen, DeepSeek, Mistral, and Llama backbones.

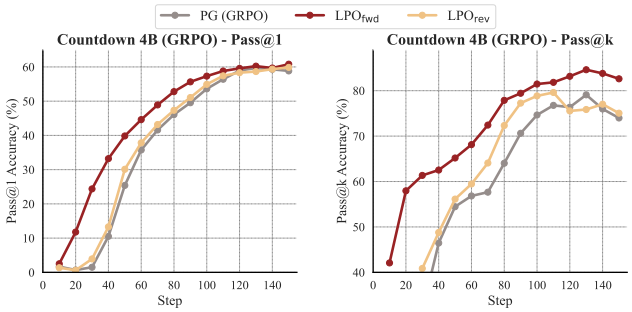


Figure 12: Empirical evaluation on the Countdown task under a fully on-policy regime (one gradient update per iteration).

all evaluated LLMs, LPO consistently improves performance on the Countdown task over the PG baseline, with especially stable gains under Pass@64 evaluation.

E.4 Fully On-Policy Optimization

To empirically validate the theoretical connections established in Proposition 1, we conduct an evaluation on Countdown under a strictly fully on-policy regime, as shown in Fig. 12. By setting both the batch size and the optimization mini-batch size to 256, we ensure exactly one gradient update is performed per iteration. As predicted by our analysis, the training curves of LPO$_{rev}$ are practically indistinguishable from standard GRPO, supporting that the group-based PG objective mathematically collapses into the exact reverse KL projection at the on-policy point. Furthermore, evaluating LPO$_{fwd}$ under this identical setup reveals its distinct exploration superiority: it demonstrates higher sample efficiency in early training stages and achieves a superior Pass@k accuracy.

E.5 Robustness Under Noisy Rewards

To further evaluate the effect of noisy rewards, we additionally simulate an unreliable verifier on Countdown by randomly flipping 10% and 50% of reward labels during training. Encouragingly, as shown in Table 4, LPO remains consistently competitive under noisy rewards, suggesting that the explicit target-projection formulation does not appear to exhibit higher sensitivity to reward noise in practice.

E.6 Evaluation Results

Table 3 presents the final evaluation on mathematics benchmarks, with k configurations following standard practices [Gao et al., 2025, Qu et al., 2025b]. Furthermore, to assess out-of-distribution (OOD)

Table 3: Evaluation on mathematics benchmarks. **Base** denotes the backbone without RLVR. **Pass@1** and **Pass@k** are computed by averaging benchmark-level Avg@k and Pass@k scores across benchmarks, respectively. **Bold** and underlined values indicate the best and second-best results for each policy gradient baseline, respectively.

Backbone	Method	MATH500	Olympiad.	Minerva.		AMC23		AIME24		AIME25		Pass@1↑	Pass@k↑
		Avg@1	Avg@1	Avg@4	pass@4	Avg@32	pass@32	Avg@32	pass@32	Avg@32	pass@32		
Qwen3-1.7B-Base	**Base**	52.8	21.2	21.2	32.8	30.0	79.3	3.4	25.0	3.3	23.8	22.0	40.2
	GRPO	71.4	33.5	29.8	37.2	45.4	83.7	10.8	26.5	4.2	20.9	32.5	42.1
	↪**LPO$_{\text{fwd}}$**	72.0	38.1	29.9	37.5	50.1	83.4	13.2	33.7	8.6	29.6	**35.3**	**46.1**
	↪**LPO$_{\text{rev}}$**	73.0	37.1	29.2	36.4	47.0	83.1	13.9	26.6	9.6	22.9	35.0	42.3
	DrGRPO	69.2	36.1	29.8	36.8	43.3	76.2	8.5	25.4	6.3	30.4	32.2	42.2
	↪**LPO$_{\text{fwd}}$**	73.8	36.5	28.3	36.5	46.2	75.9	10.3	27.1	5.3	30.4	**33.4**	42.5
	↪**LPO$_{\text{rev}}$**	70.0	36.7	28.6	38.1	45.6	78.9	10.3	31.7	6.3	26.7	32.9	**43.9**
	MaxRL	72.6	35.3	28.6	36.9	42.4	79.0	10.6	30.8	4.8	24.6	32.4	42.8
	↪**LPO$_{\text{fwd}}$**	71.8	37.5	30.5	36.4	49.9	85.5	11.8	28.6	8.5	31.9	**35.0**	**45.6**
	↪**LPO$_{\text{rev}}$**	72.6	35.8	28.8	36.3	46.1	82.6	10.7	28.3	7.7	32.6	33.6	45.0
Qwen3-8B-Base	**Base**	68.0	33.7	31.7	44.1	46.5	84.3	12.1	39.9	7.9	31.8	33.3	50.0
	GRPO	86.2	51.9	40.4	46.1	63.8	79.1	24.0	52.1	19.5	40.7	47.6	54.5
	↪**LPO$_{\text{fwd}}$**	86.4	55.8	42.3	48.3	69.1	95.1	29.3	51.0	19.1	38.7	**50.3**	**58.3**
	↪**LPO$_{\text{rev}}$**	85.0	53.9	41.1	46.9	67.0	93.1	23.3	45.7	21.6	40.2	48.7	56.5
	DrGRPO	85.8	54.7	42.2	48.4	67.7	89.7	24.9	56.3	19.3	47.0	49.1	**60.4**
	↪**LPO$_{\text{fwd}}$**	87.4	51.6	42.6	48.3	70.2	91.5	25.6	59.5	19.8	38.4	**49.5**	59.4
	↪**LPO$_{\text{rev}}$**	84.6	51.0	42.0	47.8	64.9	91.4	26.0	53.0	17.9	35.3	47.7	56.9
	MaxRL	86.4	53.6	42.6	48.9	66.0	93.4	23.9	48.6	18.9	41.7	48.6	58.2
	↪**LPO$_{\text{fwd}}$**	89.4	54.5	44.8	52.3	69.0	94.5	23.9	57.6	21.3	47.8	50.5	**63.1**
	↪**LPO$_{\text{rev}}$**	87.6	55.8	45.3	52.3	70.1	92.5	22.5	52.6	22.5	43.6	**50.6**	60.3

Table 4: Performance comparison under noisy rewards on Countdown.

Method	10% Noise		50% Noise	
	pass@1	pass@64	pass@1	pass@64
GRPO	52.58	68.91	30.35	59.15
LPO$_{\text{fwd}}$	**57.41**	**80.04**	32.73	62.96
LPO$_{\text{rev}}$	53.88	78.69	**41.31**	**66.52**

generalization, Table 5 compares LPO against counterpart PG baselines (all trained on MATH using Qwen3-8B-Base) across general reasoning tasks: MMLU-Pro Wang et al. [2024], ARC-c Clark et al. [2018], and GPQA-diamond Rein et al. [2024]. While specific LPO variants can improve the overall average, OOD evaluation exhibits inherent variance, suggesting multi-domain joint training as a natural direction for future work.

Table 5: Out-of-Distribution evaluation of LPO against baseline methods trained on MATH.

Method	ARC-c Avg@32	MMLU-Pro Avg@32	GPQA Avg@32	Avg.↑
GRPO	33.4	56.0	25.3	38.2
↪LPO	38.4	53.5	23.7	**38.5**
Dr.GRPO	33.2	55.4	23.8	37.5
↪LPO	36.4	58.5	25.6	**40.2**
MaxRL	22.5	49.8	18.7	30.3
↪LPO	26.3	51.2	19.3	**32.3**

F Data Examples

The prompt templates for MATH and Geometry3k are adopted from the official `verl` framework; the template for Countdown follows the format introduced in Pan et al. [2025], and we directly use the prompts for PRIME code from Cui et al. [2025].

MATH example

Prompt:
The points $(9, -5)$ and $(-3, -1)$ are the endpoints of a diameter of a circle. What is the sum of the coordinates of the center of the circle? Let's think step by step and output the final answer within `\boxed{}`.
Ground-Truth Answer:
0

Countdown example

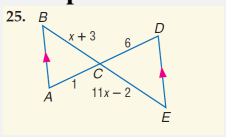
Prompt:
A conversation between User and Assistant. The user asks a question, and the Assistant solves it. The assistant first thinks about the reasoning process in the mind and then provides the user with the answer.
User: Using the numbers [2, 54, 17], create an equation that equals 35. You can use basic arithmetic operations (+, -, *, /) and each number can only be used once. Show your work in <think> $<$ /think> tags. And return the final answer in <answer> $<$ /answer> tags, for example <answer> $(1 + 2)/3$ </answer>.
Assistant: Let me solve this step by step.
<think>

PRIME-Code example

System Prompt:
When tackling complex reasoning tasks, you have access to the following actions. Use them as needed to progress through your thought process.
[ASSESS] [ADVANCE] [VERIFY] [SIMPLIFY] [SYNTHESIZE] [PIVOT] [OUTPUT]
You should strictly follow the format below:
[ACTION NAME]
Your action step 1
Your action step 2
Your action step 3
...
Next action: [NEXT ACTION NAME]
Prompt
Problem: Given a natural number N less than or equal to 12, find the smallest natural number such that the number of divisors is exactly N.
Constraints
* $1 \leq N \leq 12$
Input: One natural number N is given in one line.
Output: Output the smallest natural number on a line so that the number of divisors is exactly N.
Examples
Input 1 Output 1 Input 2 Output 2 Input 3 Output 4
Write Python code to solve the problem. Present the code in
```
python
```
```
Your code
```
at the end.

Geometry3k example

Prompt:



Find x. You FIRST think about the reasoning process as an internal monologue and then provide the final answer. The reasoning process MUST BE enclosed within <think> < /think> tags. The final answer MUST BE put in `\boxed{}`.

Ground-Truth Answer:

4

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: The paper clearly delineates the contributions and scope at the introduction.
   Guidelines:
   - The answer [N/A] 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 [N/A] 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: Please refer to Sec. 6 and additional clarifications in Appendix C.1 and Appendix C.2.
   Guidelines:
   - The answer [N/A] 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: Please refer to Appendix B for the proofs.
   Guidelines:
   - The answer [N/A] 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: Our code is available at https://anonymous.4open.science/r/LPO. The implementation details are provided in Appendix D.
   Guidelines:
   - The answer [N/A] 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: Our code is available at https://anonymous.4open.science/r/LPO.
   Guidelines:
   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.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://neurips.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) necessary to understand the results?
   Answer: [Yes]
   Justification: Please refer to Appendix D for the implementation details.
   Guidelines:
   - The answer [N/A] 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: Following standard practice in RLVR [Shao et al., 2024b, Yu et al., 2025, Qu et al., 2025b], we do not report error bars due to the high computational cost of training and evaluation across multiple seeds.
   Guidelines:
   - The answer [N/A] 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: Please refer to Appendix D for the compute resources.
Guidelines:

- The answer [N/A] 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: The research conducted in this paper conforms in every respect to the NeurIPS Code of Ethics.
Guidelines:

- The answer [N/A] 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: [Yes]
Justification: Please refer to Appendix C.8 for the discussion on broader impacts.
Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?
Answer: [N/A]
Justification: The paper doesn't release data or models that have a high risk for misuse.
Guidelines:

- The answer [N/A] 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: We properly cite the original paper and provide necessary details for the codes and tasks used in the paper.
Guidelines:

- The answer [N/A] 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: [Yes]
Justification: The codes we released are well documented and anonymized.
Guidelines:

- The answer [N/A] 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: [N/A]
Justification: The paper does not involve crowdsourcing nor research with human subjects.
Guidelines:

- The answer [N/A] 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: [N/A]
Justification: The paper does not involve crowdsourcing nor research with human subjects.
Guidelines:

- The answer [N/A] 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 rigor, or originality of the research, declaration is not required.
Answer: [Yes]
Justification: The LLM is used as the policy model in RL post-training, and we provide detailed LLM usage in Appendix D.2.
Guidelines:

- The answer [N/A] 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 in the NeurIPS handbook for what should or should not be described.