ON ADVANTAGE ESTIMATES FOR MAX@K POLICY GRADIENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning with verifiable rewards is widely used for post-training reasoning models, but sparse outcome rewards make exploration difficult. A complementary approach is to optimize inference-time objectives such as pass@K and max@K directly, yet existing policy-gradient estimators for these objectives use different signals, baselines, and normalizations, making their relationships unclear. We study this issue through baseline design and advantage centering. Starting from the advantage estimator of a leading method in the field, we show that it is policy-gradient unbiased but yields a non-centered advantage. We then introduce a Leave-Two-Out baseline that preserves policy-gradient unbiasedness while making realized batch advantages exactly centered. The resulting method, MaxPO, has an efficient quadratic-time implementation and integrates naturally into group-based RL for LLM post-training. We further derive the canonical finite-batch advantage for max@K, providing a unified view of existing advantage estimators. Empirically, we verify that the L2O baseline reduces gradient variance and outperforms non-centered alternatives.

1 INTRODUCTION

Reinforcement learning with verifiable rewards has become a common post-training method for large language models, particularly for reasoning tasks where the final answer can be checked automatically (Shao et al., 2024). In such settings, rewards are typically sparse and outcome-based: a sampled completion receives a positive reward only if its final answer is correct. This sparsity makes exploration difficult, since the policy must discover useful reasoning paths in a large combinatorial space. Existing work has addressed this issue by adding exploration incentives, including entropy regularization (Cheng et al., 2026; Zheng et al., 2025b; Cui et al., 2025) and count-based bonuses (Song et al., 2025a; Bai et al., 2025). These methods can be effective, but they introduce auxiliary objectives and require careful tuning of their trade-off with the task reward.

A complementary direction is to optimize the objective that is actually used at inference time. Reasoning models are often evaluated by repeated sampling: a prompt is considered solved if at least one of several sampled completions is correct. This motivates pass@K, the probability that at least one out of K samples is correct, and its continuous-reward generalization max@K. Several recent works have proposed unbiased policy-gradient estimators for pass@K and max@K (Koyamada et al., 2022; Tang et al., 2025; Zheng et al., 2025a; Chen et al., 2025b; Walder & Karkhanis, 2025; Peng et al., 2025; Bagirov et al., 2025; Tuyls et al., 2025). However, because the max operation couples multiple samples, the relationship among these estimators is not immediately clear. As a result, it remains unclear which estimator should be preferred, even when they target the same objective.

We study this question through a classical criterion from the policy-gradient literature: the quality of the advantage estimator. Subtracting a baseline can reduce variance while preserving unbiasedness (Williams, 1992; Greensmith et al., 2004; Mei et al., 2022), and a particularly desirable advantage estimator is centered. For batched policy-gradient estimators, an even stronger property is exact centering: the realized batch-average advantage is zero for every sampled batch. This property removes unnecessary common-mode variation in the gradient estimate and provides a principled way to compare otherwise unbiased estimators. From this perspective, we first revisit the estimator of Walder & Karkhanis (2025) from the Expected Improvement (EI) viewpoint of Nishimori et al. (2026). This estimator provides a general unbiased policy-gradient estimator for max@K with

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

$K \leq B$, where B is the number of sampled responses. We show that the PG estimator is unbiased but the resulting advantage is not centered, leaving room for variance reduction through an additional baseline. To address this, we introduce a **Leave-Two-Out (L2O)** baseline. The resulting estimator remains unbiased as a policy-gradient estimator, while its advantage is zero-mean in expectation and exactly centered within every realized batch. We also derive an efficient $O(B^2)$ vectorized implementation and incorporate the estimator into group-based reinforcement learning for LLM post-training, which we call MaxPO, or Max@K Policy Optimization.

We then step back from the EI estimator and ask what the advantage should be for a finite-batch max@K policy gradient. In ordinary policy gradients, an advantage is a return minus a baseline. For max@K, however, the relevant return of a response is not its individual reward alone, because the objective evaluates the maximum reward within a group of K samples. Thus, the natural return assigned to response i is the expected max@K value of a group conditioned on containing i, which we denote by u_i. The corresponding baseline should remove the contribution of response i while preserving the same max@K objective; this gives the leave-one-out expected max@K value v_i, computed from groups that exclude i. This leads to our canonical finite-batch advantage $u_i - v_i$: the max@K analogue of a return minus a leave-one-out baseline. This view explains why MaxPO recovers the centered canonical signal, and it also provides a common language for comparing existing estimators. Under this view, current methods fall into two categories: **uncentered** signals such as EI-only/PKPO, and **(normalized) canonical** signals such as MaxPO.

Empirically, we first validate the variance-reduction effect of our method in controlled bandit and maze environments, with the maze results reported in Appendix G.2. We also validate the practicality of our method for LLM reasoning tasks using Llama-3.2-3B-Instruct and Qwen2.5-Math-7B (Sec. 5.2). Specifically, on the Llama-3.2-3B-Instruct model, our method (EI+L2O) achieved an average reduction of 77.4% in gradient variance during training compared to PKPO, an existing method without L2O (Walder & Karkhanis, 2025). Our method also improved task-average pass@256 performance by 5.2% on Qwen2.5-Math-7B and by 2.4% on Llama-3.2-3B-Instruct relative to PKPO across five math reasoning benchmarks (AIME24, AIME25, AMC23, MATH500 (Hendrycks et al., 2021), and Minerva (Lewkowycz et al., 2022)). Furthermore, we demonstrated that our method consistently outperforms strong representative LLM post-training baselines (PKPO, GRPO, Entropy-Adv).

2 Background

In this section, we first formalize the setting of reasoning tasks and review the objectives of pass@K and max@K policy optimization. We then outline the motivation for our work and introduce the fundamental concepts required to construct our proposed estimator.

Setting. We consider a setting in which an agent generates an action $a \in \mathcal{A}$, where $\mathcal{A}$ is a finite set, and the action is evaluated by a reward function $r : \mathcal{A} \to \mathbb{R}$. Our goal is to learn a policy $\pi_\theta \in \Delta(\mathcal{A})$, parameterized by $\theta \in \mathbb{R}^d$, that maximizes the expected reward $J_{\mathrm{RL}}(\theta) = \mathbb{E}_{a\sim\pi_\theta}[r(a)]$. When it is clear from the context, we omit the dependence on π_θ for brevity. In practice, the policy may take additional information as input (e.g., questions in LLM reasoning), but we omit this dependence, and the same statements naturally apply to that setting (Sec. 3.2). During optimization, we assume access to a batch of B actions $a_1, \ldots, a_B \in \mathcal{A}$ sampled from the policy and rewards $r_1, \ldots, r_B \in \mathbb{R}$. We denote the batch by $\mathcal{D} = (a_{1:B}, r_{1:B})$.

2.1 Policy Gradient Estimation and Variance Reduction

In this study, we focus on the policy gradient (PG) method, which directly optimizes π_θ via gradient ascent using $\nabla J(\theta) = \mathbb{E}_{a\sim\pi_\theta}[r(a)\nabla_\theta \log \pi_\theta(a)]$ (Williams, 1992). Because policy gradient estimates can exhibit high variance, we introduce a constant baseline b to reduce variance (Greensmith et al., 2004). This yields the following expression.

$$\nabla J(\theta) = \mathbb{E}_{a\sim\pi_\theta}\left[(r(a) - b)\nabla_\theta \log \pi_\theta(a)\right]. \tag{1}$$

We use the term *advantage* to denote the action-dependent scalar multiplier of the score function (i.e., $r(a) - b$ in Eq. (1)). As long as the baseline is action-independent, the PG estimator remains unbiased because $\mathbb{E}_{a\sim\pi_\theta}[b\,\nabla_\theta \log \pi_\theta(a)] = 0$. A common choice is to use the expected reward as the baseline, $b = \mathbb{E}_{a\sim\pi_\theta}[r(a)]$, so that the advantage has zero mean: $\mathbb{E}_{a\sim\pi_\theta}[r(a) - b] = 0$ (Greensmith

et al., 2004). Using such a zero-mean advantage removes a constant offset from the reward signal, which is often a dominant source of variance.

Batch estimation. Given the batch $\mathcal{D}$, we can approximate the gradient using the REINFORCE estimator (Williams, 1992) $\hat{g} = \frac{1}{B}\sum_{i=1}^{B}(r_i - b_i)\nabla_\theta \log \pi_\theta(a_i)$. Here, b_i is the baseline for sample i. We distinguish two notions of unbiasedness and one notion of centering:

1. **Unbiased PG estimator:** We call a PG estimator $\hat{g}$ unbiased when $\mathbb{E}_{\mathcal{D}\sim\pi_\theta}[\hat{g}] = \nabla J(\theta)$. Here, $\mathbb{E}_{\mathcal{D}\sim\pi_\theta}[\cdot]$ denotes expectation over random batches $\mathcal{D} = (a_{1:B}, r_{1:B})$ obtained by i.i.d. sampling from π_θ and evaluating rewards under r. This requires that the baseline b_i does not depend on r_i, which would introduce action dependence into the PG estimator.
2. **Unbiased advantage:** We call an advantage estimator (e.g., $r_i - b_i$) unbiased when $\mathbb{E}_{\mathcal{D}\sim\pi_\theta}[r_i - b_i] = 0$.
3. **Centering:** We call the realized batch-average advantage centered when $\frac{1}{B}\sum_{i=1}^{B}(r_i - b_i) = 0$. This is a stronger per-batch property than unbiasedness in expectation; the latter does not in general imply centering.

Unbiasedness of the PG estimator is a prerequisite for policy gradient methods, and an unbiased advantage is crucial for variance reduction (Greensmith et al., 2004). Therefore, we aim to achieve both forms of unbiasedness, and it turns out that centering also holds. For a batch PG estimator, a naïve way to obtain an unbiased advantage estimator is to set the baseline to the batch mean reward, $b_i = \frac{1}{B}\sum_{j=1}^{B} r_j$, as an estimator of $\mathbb{E}_{a\sim\pi_\theta}[r(a)]$. However, this violates the independence condition because b_i depends on r_i, resulting in a biased PG estimator. A common remedy is to use a Leave-One-Out (L1O) baseline, $b_i = \frac{1}{B-1}\sum_{j\neq i} r_j$, to avoid dependence on r_i (Parmas et al., 2018; Mnih et al., 2016). Although L1O works in this simple RL setting, it may fail for more complicated objectives, as is the case for max@K (see Sec. 3.1).

2.2 Pass@K and max@K Policy Optimization

Recently, in the context of RL for reasoning tasks with binary rewards (correct/incorrect), directly optimizing the pass@K metric has been proposed to incentivize answer diversity (Tang et al., 2025; Zheng et al., 2025a; Chen et al., 2025b; Walder & Karkhanis, 2025; Peng et al., 2025; Bagirov et al., 2025; Tuyls et al., 2025). Pass@K optimization aims to maximize the probability that at least one of K sampled answers is correct. The max@K objective generalizes pass@K to continuous reward functions (Walder & Karkhanis, 2025) and is defined as the expected maximum reward among $K \leq B$ samples.

$$J^K(\theta) := \mathbb{E}_{a_{1:K}\sim\pi_\theta}\left[\max_{k=1,\ldots,K} r(a_k)\right]. \tag{2}$$

The PG for this objective was first given by Koyamada et al. (2022):

$$\nabla J^K(\theta) := \mathbb{E}_{a_{1:K}}\left[\max_{k=1,\ldots,K} r(a_k)\sum_{k=1}^{K}\nabla_\theta \log \pi_\theta(a_k)\right]. \tag{3}$$

Our goal is to propose an estimator that is unbiased as a PG estimator and also yields an advantage with zero expectation. However, the formulation in Eq. (3) makes it difficult to evaluate the advantage because the K actions are coupled through the max operator. To address this issue, we leverage an alternative formulation of the max@K PG.

Expected Improvement Formulation. Nishimori et al. (2026) proposed a method to decouple the max@K PG into a per-action expectation by building on the baseline proposed by Tang et al. (2025): $W_{-k} := \max_{k'\neq k} r(a_{k'})$. Using the identity $\max_k r_k - W_{-k} = (r_k - W_{-k})_+$, where $(z)_+ := \max(z, 0)$, the PG can be rewritten as:

$$\nabla J^K(\theta) = \mathbb{E}\left[\sum_{k=1}^{K}\nabla_\theta \log \pi_\theta(a_k)\left(r(a_k) - W_{-k}\right)_+\right].$$

By exploiting the symmetry under the i.i.d. assumption, we arrive at the simplified form (Prop. 2 of Nishimori et al. (2026)):

$$\nabla J^K(\theta) = K\mathbb{E}_{a\sim\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a)\, s(a)\right], \quad s(a) := \mathbb{E}_{a_{1:K-1}\sim\pi_\theta}\left[\left(r(a) - W_{K-1}\right)_+\right], \tag{4}$$

with $W_{K-1} = \max_{j=1,\dots,K-1} r(a_j)$. $s(a)$ is called the **Expected Improvement (EI)** of action a, as it quantifies the expected gain of a over the best of $K-1$ other samples (Nishimori et al., 2026).

Motivation for Further Variance Reduction. Compared to the standard PG in RL (Eq. (1)), the EI term $s(a)$ serves as the advantage. Moreover, $s(a)$ corresponds to the expected value of the advantage in the $\max@K - \max@(K-1)$ estimator proposed by Walder & Karkhanis (2025). While their estimator reduces variance compared to naïve approaches, Eq. (4) reveals a critical limitation. Because $s(a)$ is the output of a ReLU, it is non-negative, implying $\mathbb{E}_{a\sim\pi_\theta}[s(a)] > 0$. This consistently positive advantage corresponds to an underestimated baseline, which can be particularly harmful because it may steer policy optimization toward suboptimal actions (Chung et al., 2021). Thus, **their estimator is unbiased as a PG estimator but not as an advantage estimator**, leaving room for further variance reduction through appropriate baseline subtraction.

3 Proposed Method

In Sec. 3.1, we derive a baseline that **guarantees unbiasedness of both the PG estimator and the resulting advantage and exactly centers the realized batch-average advantage**, and we present an efficient computation method. Finally, Sec. 3.2 describes how to integrate our approach into modern RL algorithms, instantiating it as MaxPO (Max@K Policy Optimization).

3.1 Unbiased Advantage Estimation

Given a batch $\mathcal{D} = (a_{1:B}, r_{1:B})$, we first construct unbiased estimators s_i of the expected improvement $s(a_i)$ defined in Eq. (4).

$$s_i := \mathbb{E}_{\mathcal{I}}\left[\left(r_i - \max_{j\in\mathcal{I}} r_j\right)_+\right]. \tag{5}$$

Here, $\mathcal{I}$ is a subset of size $K-1$ drawn uniformly *without replacement* from the indices $\{1,\dots,B\}\setminus\{i\}$. By U-statistics theory (Hoeffding, 1992), s_i is an unbiased estimator of $s(a_i)$. Note that s_i is equivalent to one of Walder & Karkhanis (2025)'s advantage estimators (max@K - max@(K-1)).

Recall the three notions introduced in Sec. 2.1. For the desired estimator, we seek a baseline b_i that satisfies the following three conditions: (1) **Independence:** b_i does not depend on r_i, ensuring an unbiased PG estimator; (2) **Unbiased advantage:** $\mathbb{E}_{\mathcal{D}\sim\pi_\theta}[s_i - b_i] = 0$; and (3) **Centering:** $\frac{1}{B}\sum_{i=1}^{B}(s_i - b_i) = 0$ for every realized batch. As discussed in Sec. 2.1, a common approach is the L1O baseline, $b_{-i}^{\mathrm{L1O}} := \frac{1}{B-1}\sum_{j\neq i} s_j$. However, in the max@K setting, the estimator $s_j, j\neq i$ can implicitly depend on r_i, because r_i acts as a potential comparator (maximum over the subset of size $K-1$) in the construction of s_j. Consequently, b_{-i}^{L1O} can correlate with r_i, thereby violating the independence condition and leading to a biased PG estimator.

Leave-Two-Out (L2O) Baseline. To satisfy all three conditions, we propose the *Leave-Two-Out* (L2O) baseline:

$$b_{-i}^{\mathrm{L2O}} := \frac{1}{B-1}\sum_{j\neq i} s_j^{(-i)}, \quad s_j^{(-i)} := \mathbb{E}_{\mathcal{I}'}\left[\left(r_j - \max_{k\in\mathcal{I}'} r_k\right)_+\right]. \tag{6}$$

Here, $\mathcal{I}'$ is a subset of size $K-1$ drawn uniformly from $\{1,\dots,B\}\setminus\{i,j\}$. We exclude i to ensure independence from r_i, while keeping the comparator subset size $K-1$ consistent with the original EI definition. This requires $K-1 \le B-2$, i.e., $K \le B-1$. By construction, b_{-i}^{L2O} is independent of r_i because r_i does not appear as a comparator in its computation. The following proposition establishes the centering condition. Together with the independence property above, it implies that the full estimator $\frac{K}{B}\sum_{i=1}^{B}\nabla_\theta \log \pi_\theta(a_i)(s_i - b_{-i}^{\mathrm{L2O}})$ is an unbiased PG estimator for the max@K objective, and that $s_i - b_{-i}^{\mathrm{L2O}}$ is an unbiased advantage estimator.

Algorithm 1 Max@K Policy Optimization

Input: policy π_θ, objective size K, number of questions per batch M, group size G with $2 \le K \le G-1$.
while not converged **do**
  Sample questions $\{x_m\}_{m=1}^{M}$.
  for each question x_m **do**
    Sample G outputs $\{a_j^m\}_{j=1}^{G} \sim \pi_\theta(\cdot \mid x_m)$ and evaluate rewards $\{r_j^m\}_{j=1}^{G}$.
    Compute EI scores $\mathbf{s}^m$ and L2O baselines $\mathbf{b}^{m,\mathrm{L2O}}$.
    Set sequence-level advantages $\mathbf{A}^m = \mathbf{s}^m - \mathbf{b}^{m,\mathrm{L2O}}$.
  end for
  Update θ using a group-based policy-gradient objective with advantages $\{\mathbf{A}^m\}_{m=1}^{M}$.
end while

Proposition 3.1 (Unbiasedness and Centering of the L2O Baseline)**.** *For any i, the L2O advantage satisfies*

$$\mathbb{E}_{\mathcal{D}\sim\pi_\theta}\left[s_i - b_{-i}^{\mathrm{L2O}}\right] = 0, \quad \textit{and} \quad \frac{1}{B}\sum_{i=1}^{B}\left(s_i - b_{-i}^{\mathrm{L2O}}\right) = 0, \tag{7}$$

for any realized batch $\mathcal{D}$. Hence $s_i - b_{-i}^{\mathrm{L2O}}$ is unbiased as an advantage estimator and exactly centered at the batch level.

The proof is in Appendix D.1. While the theoretical properties of the L2O baseline are desirable, its practical application hinges on computational efficiency. Computing $\mathbf{b}^{\mathrm{L2O}} := [b_{-1}^{\mathrm{L2O}}, \ldots, b_{-B}^{\mathrm{L2O}}]^\top$ naively costs $O(B^3)$ because it evaluates $O(B)$ EI terms for each of the B indices i, and each EI term naively scans $O(B)$ candidate comparators in the (reduced) batch. However, leveraging the ReLU-based formulation of the EI estimator, we can compute the L2O baseline in $O(B^2)$.

Theorem 3.2 (Efficient Computation of the L2O Baseline)**.** *Given a batch of rewards $\mathbf{r} = [r_1, \ldots, r_B]^\top$, the L2O baseline $\mathbf{b}^{\mathrm{L2O}}$ can be computed in $O(B^2)$ time.*

The proof is provided in Appendix E. The resulting vectorized formulation maps directly to GPU-friendly dense primitives (e.g., broadcasted elementwise operations and matrix–vector products), making it straightforward to leverage GPU parallelism in practice.

3.2 Integration with RL Algorithms for LLM Training

We integrate L2O-based advantage estimation into group-based RL for LLM post-training as **MaxPO** (Max@K Policy Optimization). Algorithm 1 summarizes the procedure. For each question x_m, we sample G outputs $\{a_j^m\}_{j=1}^{G}$ from $\pi_\theta(\cdot \mid x_m)$ and evaluate their sequence-level rewards $\{r_j^m\}_{j=1}^{G}$. Within each question-specific group, G plays the role of the batch size B used in the EI and L2O computations. We compute group-wise EI scores $\mathbf{s}^m$ and L2O baselines $\mathbf{b}^{m,\mathrm{L2O}}$, and define the sequence-level advantage $A_j^m = s_j^m - b_j^{m,\mathrm{L2O}}$. Before clipping and regularization, the corresponding group-wise estimator is $\frac{K}{G}\sum_{j=1}^{G}\nabla_\theta \log \pi_\theta(a_j^m \mid x_m) A_j^m$, which is an unbiased policy-gradient estimator for the max@K objective within the group. MaxPO is applicable to any policy-gradient algorithm that accepts sequence-level advantages. In our LLM experiments, we instantiate it with group-based training (Shao et al., 2024; Liu et al., 2025). Since verifier rewards are assigned at the sequence level, we broadcast A_j^m to every token in output a_j^m when forming the token-level loss. The resulting objective is identical to standard group-based clipped policy optimization except for the advantage construction. EI-only methods use the raw EI scores $\mathbf{s}^m$, whereas MaxPO subtracts the L2O baseline and uses the centered advantages $\mathbf{s}^m - \mathbf{b}^{m,\mathrm{L2O}}$. We give the exact token-level clipped objective and KL regularization term in Appendix C.

4 Canonical Finite-Batch Advantage for Max@K Policy Gradient

Sec. 3 derived MaxPO by adding an L2O baseline to the EI estimator of Walder & Karkhanis (2025), yielding an unbiased policy-gradient estimator with a centered advantage for max@K. However, EI is

Table 1: Baseline view of pass@K/max@K advantages. ✓ indicates methods used in the original papers' experiments (for MaxPO, this refers to our experiments). Fixed constants are shown when they clarify equivalence up to learning-rate scaling for fixed B and K. The methods used in the experiments fall into two categories: (1) Uncentered, which does not center the signal, and (2) (Normalized) Canonical, which is canonical up to a fixed scale.

Used	Method	Mathematical signal	Relation to canonical
✓	Walder & Karkhanis (2025), `EI-only`	$s_i = u_i - w_i$	Uncentered
✓	MaxPO	$s_i - b_i^{\mathrm{L2O}} = u_i - v_i$	Canonical.
✓	Chen et al. (2025b), Eq. (14,15)	$\frac{B-K}{B}(u_i - v_i)/\sigma_{\mathrm{group}}$	Normalized. For pass@K
	Chen et al. (2025b), w.o. std.	$\frac{B-K}{B}(u_i - v_i)$	Proportional. For pass@K
	Bagirov et al. (2025), Eq. (9)	$\tilde{r}_i = \frac{K}{B}u_i$	Proportional to u_i
	Bagirov et al. (2025), mean-centered	$\tilde{r}_i - \bar{\tilde{r}} = \frac{K(B-K)}{B^2}(u_i - v_i)$	Proportional
✓	Bagirov et al. (2025), BoN mean, Appendix	$\frac{K(B-K)}{B^2}(u_i - v_i)/\mathrm{std}(\tilde{r})$	Normalized

only one possible starting point for constructing pass@K/max@K advantages. Recent methods use different primitives, including analytical pass@K signals, all-subsets reward transformations, and standard-deviation-normalized advantages (Chen et al., 2025b; Bagirov et al., 2025; Tang et al., 2025). These differences make direct comparison difficult: the estimators may center different vectors, subtract different baselines, or apply different batch-dependent normalizations.

This section provides a unified baseline view of these estimators. We first propose a canonical finite-batch advantage form based on the leave-one-out principle (Parmas et al., 2018; Mnih et al., 2016). We then show that existing estimators can be understood as uncentered signals, fixed-scale versions of this canonical direction, or normalized variants of it.

4.1 Canonical Finite-Batch Marginal Advantage

For a fixed prompt, let $\mathcal{B} = \{1, \dots, B\}$ denote the sampled responses and define $M(S) := \max_{j \in S} r_j$ for any subset $S \subseteq \mathcal{B}$. We define

$$u_i := \frac{1}{\binom{B-1}{K-1}} \sum_{\substack{S \subseteq \mathcal{B}\setminus\{i\} \\ |S|=K-1}} M(S \cup \{i\}), \qquad v_i := \frac{1}{\binom{B-1}{K}} \sum_{\substack{T \subseteq \mathcal{B}\setminus\{i\} \\ |T|=K}} M(T). \tag{8}$$

Here, u_i is the conditional expected max@K value of a group *containing* response i. It is the max@K analogue of the return of action i. The quantity v_i is the leave-one-out expected max@K value computed *without* response i. It is the corresponding leave-one-out baseline. The signal $u_i - v_i$ is therefore the finite-batch max@K analogue of a return minus a leave-one-out baseline. Because v_i excludes response i, subtracting it preserves policy-gradient unbiasedness. Moreover, a finite-subset counting argument shows that $u_i - v_i$ is exactly centered, i.e., $\sum_{i=1}^{B}(u_i - v_i) = 0$ for every realized batch (Appendix F.1). Thus, $u_i - v_i$ is the canonical centered finite-batch advantage.

4.2 Existing Estimators as Baseline Choices

EI-based advantage estimators. To connect EI with the marginal view, define $w_i := \frac{1}{\binom{B-1}{K-1}} \sum_{\substack{S \subseteq \mathcal{B}\setminus\{i\} \\ |S|=K-1}} M(S)$ the expected maximum of the $K-1$ comparator responses excluding i. The EI-only estimator can then be written as $s_i = u_i - w_i$, because $M(S \cup \{i\}) = M(S) + (r_i - M(S))_+$. Thus, EI subtracts the comparator-max baseline w_i rather than the leave-one-out max@K baseline v_i. This is sufficient for policy-gradient unbiasedness, because w_i excludes response i, but it is not the canonical centered advantage. In general, $u_i - w_i$ is nonnegative and non-centered. MaxPO corrects this difference via the L2O baseline: for any $i \in \mathcal{B}$, the L2O baseline equals $b_i^{\mathrm{L2O}} = v_i - w_i$, and therefore $s_i - b_i^{\mathrm{L2O}} = u_i - v_i$ (see Appendix F.2 for the proof). This equality shows that L2O rectifies the EI-based advantage estimator to the canonical centered advantage.

Generality of L2O. This cancellation is not specific to choosing EI as the starting signal. Suppose the starting signal for response i is an exact average over groups containing i of $M(S \cup \{i\}) - C(S)$,

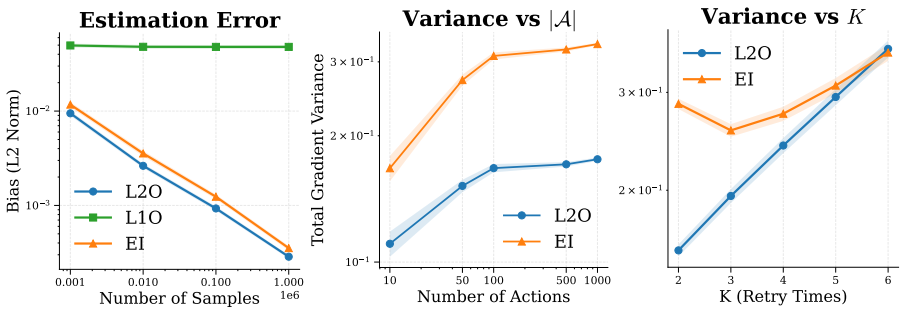


Figure 1: Estimation error (left), variance vs. action space size (center), and variance vs. K (right). Mean and standard error over 100 random seeds.

where $C(S)$ is any comparator-only baseline depending on the other $K-1$ responses S but not on response i. Then applying the same L2O construction cancels the averaged $C(S)$ term and again yields $u_i - v_i$. Thus, L2O maps any exact comparator-only baseline of this form to the canonical centered finite-batch advantage; Appendix F.3 gives the proof.

Non-EI estimators. The same view clarifies how non-EI estimators (Chen et al., 2025b; Bagirov et al., 2025) relate to MaxPO. Table 1 summarizes how existing estimators defined with $K \leq B$ relate to the canonical direction. The table suggests that the estimators used in their experiments (Walder & Karkhanis, 2025; Chen et al., 2025b; Bagirov et al., 2025) fall into two classes: *uncentered* and *(normalized) canonical*. The first class is *uncentered*: PKPO/EI-only is policy-gradient unbiased but uses the non-centered signal $u_i - w_i$. The second class is *(normalized) canonical*, or canonical up to a fixed scale: MaxPO exactly recovers $u_i - v_i$, while raw analytical pass@K and mean-centered all-subsets estimators recover the same direction up to fixed constants. Thus, for the analytical signal of Chen et al. and the mean-centered all-subsets signal of Bagirov et al., what is biased relative to MaxPO is the magnitude, not the direction; in the standard-deviation-normalized versions used in their experiments, this fixed positive scale is normalized away or absorbed into the effective step size, so it does not change the normalized update direction. This view reveals that although the derivations are different, some estimators are equivalent to MaxPO up to a fixed scale.

5 Experiments

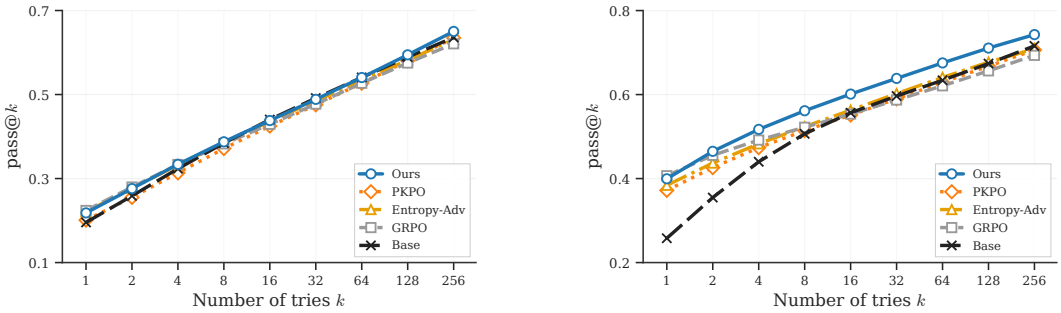
In Sec. 5.1, we validate the theoretical properties of the L2O baseline in bandits. In Sec. 5.2, we validate the efficacy of MaxPO on reasoning tasks.

5.1 Bandits

Here, we validate the two theoretical properties of the L2O baseline: (1) unbiasedness as a PG estimator and (2) variance reduction over the raw EI estimator due to centering the advantage.

Setting. To emulate a discrete-action setting such as LLM training, we consider a multi-armed bandit with rewards $\mathbf{r} \in \mathbb{R}^{|\mathcal{A}|}$ and logits $\mathbf{l} \in \mathbb{R}^{|\mathcal{A}|}$ each sampled from $\mathcal{N}(0,1)$. We sample a batch of B actions $a_i \sim \mathrm{softmax}(\mathbf{l})$, observe rewards r_i, and construct three PG estimators: (1) EI: s_i, (2) EI+L2O: $s_i - b_{-i}^{\mathrm{L2O}}$, and (3) EI+L1O: $s_i - b_{-i}^{\mathrm{L1O}}$, as defined in Sec. 3. The ground-truth gradient g_{true} is computed analytically via the closed-form derivative of the expected improvement (Proposition 1 of Nishimori et al. (2026)). For the bias plot, we vary the number of batches $N \in \{10^3, 10^4, 10^5, 10^6\}$ and measure the estimation error $\|\frac{1}{N}\sum_{j=1}^{N} \hat{g}_j - g_{\text{true}}\|$ (fixing $B=8$, $K=2$). For variance, we report the empirical total variance $\frac{1}{N}\sum_{j=1}^{N} \|\hat{g}_j - \bar{g}\|^2$ with $\bar{g} = \frac{1}{N}\sum_{j=1}^{N} \hat{g}_j$ and $N = 10^5$, sweeping (i) the action space size $|\mathcal{A}| \in \{10, 50, 100, 1000\}$ at $K=2, B=8$, and (ii) the comparator size $K \in \{2, \ldots, 6\}$ at $|\mathcal{A}| = 100, B = 8$. Detailed protocols and results for additional batch sizes are reported in Appendix G.1.

Results. The results in Fig. 1 validate our theoretical claims. First, the estimation errors of EI+L2O and EI both decrease at the theoretical $\mathcal{O}(1/\sqrt{N})$ rate (left), confirming that they are unbiased PG



(a) Llama-3.2-3B-Instruct (b) Qwen2.5-Math-7B

Figure 2: Task-average pass@k ($k \leq 256$). Unweighted average over AIME24, AIME25, AMC23, MATH500, and Minerva (temperature 0.6, top-p 0.95). Our method demonstrates consistent improvement over strong baselines.

estimators, whereas EI+L1O exhibits an approximately constant error, confirming that L1O yields a biased PG. Regarding variance, the variance-reduction effect of L2O over raw EI grows with the action space size (center), a regime particularly relevant for LLMs. Furthermore, L2O is most effective for moderate K relative to $B = 8$ (right): the L2O baseline must form $K - 1$ comparators from $B - 2$ samples to preserve unbiasedness, so a too-small comparator budget yields a noisy baseline and diminishes the variance-reduction effect.

5.2 LLM Reasoning Experiments

We evaluate the efficacy of our variance-reduced objective on challenging math reasoning tasks. Our experiments address two questions: (i) whether our estimator improves pass@k on standard reasoning benchmarks, and (ii) whether it stabilizes RL optimization by reducing the variance of the gradient estimator during training.

5.2.1 Experimental Setting

Training. We perform RL fine-tuning on Llama-3.2-3B-Instruct (Grattafiori et al., 2024) and Qwen2.5-Math-7B (Yang et al., 2024). We compare our method (EI+L2O) against three baselines: GRPO (Shao et al., 2024), Entropy-Adv (Cheng et al., 2026), and PKPO (Walder & Karkhanis, 2025). Our method follows the algorithm described in Sec. 3.2, and we implement the LLM RL training pipeline using the `verl` framework (Sheng et al., 2024). We include PKPO as the closest prior method that directly optimizes pass@K/max@K using an unbiased expected-improvement (EI) policy-gradient estimator; in our notation, PKPO can be viewed as the EI-only variant, so this comparison isolates the benefit of our centered advantage estimation. Note that the concurrent work of Bagirov et al. (2025) and prior work of Chen et al. (2025b) are equivalent to MaxPO up to a constant rescaling factor and standard deviation normalization (which removes any effect from rescaling). Therefore, these works also provide experimental evidence for the overall efficacy of the approach. Our contribution relative to the literature is the canonical $u_i - v_i$ unbiased form of the gradient estimator, thus our experiments focus on how this form improves over the non-centered PKPO variant. Unless otherwise stated, we use a fixed training objective size $K = 2$. Training data and hyperparameters are provided in App. H.1. Throughout this section, K refers to the training objective size (max@K/pass@K objective), while k denotes the evaluation compute in pass@k.

Evaluation. We evaluate on five math reasoning benchmarks: AIME24, AIME25, AMC23, MATH500 (Hendrycks et al., 2021), and Minerva (Lewkowycz et al., 2022). All evaluations use nucleus sampling with temperature 0.6 and top-p 0.95. To reduce evaluation variance, we generate $n = 1024$ samples for every benchmark. We report the unbiased pass@k (Chen et al., 2021) metric for $k \in \{1, 2, 4, 8, \ldots\}$, computed as pass@$k := \mathbb{E}_{x\sim\mathcal{D}}\left[1 - \binom{n-c}{k} / \binom{n}{k}\right]$, where n is the number of sampled completions and c is the number of correct completions among them. For consistent task-averaging in the main text, we report results up to $k \leq 256$ for all benchmarks, and additionally report $k \leq 1024$ for AIME24, AIME25, and AMC23 in Appendix H.2.

5.2.2 Main Results

Figure 2 shows the task-average pass@k curves up to $k = 256$ for both Llama and Qwen. The main pattern is that EI+L2O becomes strongest as inference compute increases: although GRPO is competitive at very small k, our method overtakes the baselines at moderate-to-large k, which is the regime most relevant to pass@K optimization. Compared to PKPO, our method improves task-average pass@256 by 5.2% on Qwen2.5-Math-7B and by 2.4% on Llama-3.2-3B-Instruct relative to PKPO across five math reasoning benchmarks. These gains persist over a wide range of k, supporting the claim that centering the advantage helps the policy make better use of additional test-time samples. We provide extended results up to $k = 1024$ for AIME24, AIME25, and AMC23 in Appendix H.2. For an ablation over different K values ($K \in \{2, 4, 6\}$), refer to Appendix H.2.1. Beyond mathematical reasoning, MaxPO also outperforms all baselines at every k on code generation with continuous rewards, while reducing the gradient-variance proxy (Sec. 5.2.3) by 29.3% relative to PKPO (Appendix H.3).

5.2.3 Variance Reduction During Training

To test our theoretical motivation, we measure a smoothed proxy for the variance of model gradients during RL training on Llama-3.2-3B-Instruct. Directly computing gradient variance is expensive at LLM scale, so we estimate this proxy from the states of the Adam optimizer (Kingma & Ba, 2017). Specifically, Adam maintains exponential moving averages of the first and second moments of gradients; after bias correction, we estimate a smoothed proxy for the element-wise variance as $\mathrm{Var}(g) \approx \hat{v}_t - \hat{m}_t^2$, and aggregate it across all parameters by summing. This is not the exact instantaneous gradient variance, but it is practical at LLM scale and computed identically for all methods, making it suitable for comparison.

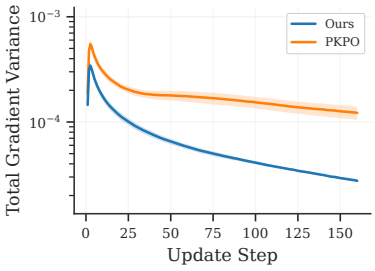


Figure 3: Adam-moment variance proxy during training (3 seeds). Our method (EI+L2O) reduces this variance proxy compared to PKPO across training, supporting the variance-reduction mechanism.

Figure 3 reports mean ± standard deviation over three random seeds. Our method yields consistently lower values of this Adam-moment variance proxy than PKPO throughout training, indicating more stable policy gradient updates. Notably, at the end of training, our method achieved a 77.4% relative reduction in this proxy compared to PKPO. This provides empirical evidence that centering the EI signal with L2O reduces estimator variance in practice, matching the paper's main theoretical claim.

6 Conclusion

In this paper, we revisit pass@K and max@K policy optimization through the lens of advantage estimation, with a focus on unbiased advantage estimation. We first identified that the advantage estimator of Walder & Karkhanis (2025), a leading method in the field, is unbiased as a PG estimator but not centered as an advantage estimator. We then introduced a Leave-Two-Out (L2O) baseline that preserves policy-gradient unbiasedness while making the realized batch advantages exactly centered. We also derived the canonical finite-batch advantage form that enables us to organize and relate existing estimators in a unified manner, classifying them into uncentered, canonical, and normalized canonical classes. Finally, we empirically verified that the L2O baseline reduces gradient variance and outperforms non-centered alternatives.

Limitations and Future Work. Our analysis focuses on finite-batch, on-policy estimation under i.i.d. sampling within each group. Extending the same canonical view to off-policy training, adaptive sampling, and settings with correlated generations is an important direction for future work. Our L2O construction also requires $K \leq B - 1$, which may limit its direct use when the optimization objective size approaches the group size. In addition, our LLM experiments focus on mathematical reasoning tasks with verifiable sequence-level rewards. Further experiments on broader domains, noisy or partial rewards, larger models, and different verifier designs are needed to understand the generality of the method.

AI USE STATEMENT

LLMs were used to edit the manuscript and to assist with algebraic manipulations in derivations proposed by the authors. The key research ideas were proposed by the authors.

REFERENCES

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2):235–256, 2002.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Farid Bagirov, Mikhail Arkhipov, Ksenia Sycheva, Evgeniy Glukhov, and Egor Bogomolov. The best of N worlds: Aligning reinforcement learning with best-of-N sampling via max@k optimisation. *arXiv preprint arXiv:2510.23393*, 2025.

Ryan Bahlous-Boldi, Isha Puri, Idan Shenfeld, Akarsh Kumar, Mehul Damani, Sebastian Risi, Omar Khattab, Zhang-Wei Hong, and Pulkit Agrawal. Vector Policy Optimization: Training for Diversity Improves Test-Time Search. *arXiv preprint arXiv:2605.22817*, 2026. doi: 10.48550/ARXIV.2605.22817. URL https://arxiv.org/abs/2605.22817.

Chenjia Bai, Yang Zhang, Shuang Qiu, Qiaosheng Zhang, Kang Xu, and Xuelong Li. Online preference alignment for language models via count-based exploration. *arXiv preprint arXiv:2501.12735*, 2025.

Dake Bu, Wei Huang, Andi Han, Atsushi Nitanda, Bo Xue, Qingfu Zhang, Hau-San Wong, and Taiji Suzuki. Consistency is not always correct: Towards understanding the role of exploration in post-training reasoning. *arXiv preprint arXiv:2511.07368*, 2025.

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018.

Fan Chen, Audrey Huang, Noah Golowich, Sadhika Malladi, Adam Block, Jordan T. Ash, Akshay Krishnamurthy, and Dylan J. Foster. The coverage principle: How pre-training enables post-training. *arXiv preprint arXiv:2510.15020*, 2025a.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, and Greg Brockman et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

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

Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Xin Zhao, Zhenliang Zhang, and Furu Wei. Reasoning with exploration: An entropy perspective. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(36):30377–30385, 2026. doi: 10.1609/aaai.v40i36.40290.

Yinlam Chow, Guy Tennenholtz, Izzeddin Gur, Vincent Zhuang, Bo Dai, Aviral Kumar, Rishabh Agarwal, Sridhar Thiagarajan, Craig Boutilier, and Aleksandra Faust. Inference-Aware Fine-Tuning for Best-of-N Sampling in Large Language Models. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/c40bed606c51c8e827c1ba75aa2da054-Paper-Conference.pdf.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Wesley Chung, Valentin Thomas, Marlos C Machado, and Nicolas Le Roux. Beyond variance reduction: Understanding the true impact of baselines on policy optimization. In *International conference on machine learning*, pp. 1999–2009. PMLR, 2021.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, et al. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025.

Xingyu Dang, Christina Baek, Kaiyue Wen, Zico Kolter, and Aditi Raghunathan. Weight ensembling improves reasoning in language models. In *Second Conference on Language Modeling*, 2025.

Jingtong Gao, Ling Pan, Yejing Wang, Rui Zhong, Chi Lu, Qingpeng Cai, Peng Jiang, and Xiangyu Zhao. Navigate the unknown: Enhancing LLM reasoning with intrinsic motivation guided exploration. *arXiv preprint arXiv:2505.17621*, 2025.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, and Alex Vaughan et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Evan Greensmith, Peter L Bartlett, and Jonathan Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5(Nov):1471–1530, 2004.

Shixiang Gu, Sergey Levine, Ilya Sutskever, and Andriy Mnih. Muprop: Unbiased backpropagation for stochastic neural networks. *arXiv preprint arXiv:1511.05176*, 2015.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, and Xiao Bi et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Anthony GX-Chen, Ankit Anand, Gheorghe Comanici, Zaheer Abbas, Eser Aygün, David Smalling, Shibl Mourad, Doina Precup, André Barreto, and Mark Rowland. Using Reward Uncertainty to Induce Diverse Behaviour in Reinforcement Learning. *arXiv preprint arXiv:2606.03962*, 2026. doi: 10.48550/ARXIV.2606.03962. URL https://arxiv.org/abs/2606.03962.

Andre He, Daniel Fried, and Sean Welleck. Rewarding the unlikely: Lifting GRPO beyond distribution sharpening. *arXiv preprint arXiv:2506.02355*, 2025.

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.

Wassily Hoeffding. A class of statistics with asymptotically normal distribution. In *Breakthroughs in statistics: Foundations and basic theory*, pp. 308–334. Springer, 1992.

Sihan Hu, Xiansheng Cai, Yuan Huang, Zhiyuan Yao, Linfeng Zhang, Pan Zhang, Youjin Deng, and Kun Chen. Emergent slow thinking in LLMs as inverse tree freezing. *arXiv preprint arXiv:2509.23629*, 2025.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. In *International Conference on Learning Representations*, 2025.

Yuxian Jiang, Yafu Li, Guanxu Chen, Dongrui Liu, Yu Cheng, and Jing Shao. Rethinking entropy regularization in large reasoning models. *arXiv preprint arXiv:2509.25133*, 2025.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2017.

Sotetsu Koyamada, Paavo Parmas, Tadashi Kozuno, and Shin Ishii. Emergence of exploration in policy gradient reinforcement learning via resetting, 2022. URL https://openreview.net/forum?id=GKsNIC_mQRG.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James Validad Miranda, Alisa Liu, Nouha Dziri, Xinxi Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Christopher Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tulu 3: Pushing frontiers in open language model post-training. In *Second Conference on Language Modeling*, 2025.

Aitor Lewkowycz, Anders Johan Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Venkatesh Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems*, 2022.

Chenyi Li, Yuan Zhang, Bo Wang, Guoqing Ma, Wei Tang, Haoyang Huang, and Nan Duan. SetPO: Set-Level Policy Optimization for Diversity-Preserving LLM Reasoning. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2602.01062.

Tianjian Li, Yiming Zhang, Ping Yu, Swarnadeep Saha, Daniel Khashabi, Jason Weston, Jack Lanchantin, and Tianlu Wang. Jointly reinforcing diversity and quality in language model generations. *arXiv preprint arXiv:2509.02534*, 2025.

Zhenwen Liang, Sidi Lu, Wenhao Yu, Kishan Panaganti, Yujun Zhou, Haitao Mi, and Dong Yu. Can LLMs guide their own exploration? gradient-guided reinforcement learning for LLM reasoning. *arXiv preprint arXiv:2512.15687*, 2025.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by ChatGPT really correct? rigorous evaluation of large language models for code generation. In *Advances in Neural Information Processing Systems*, 2023.

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. In *Conference on Language Modeling (COLM)*, 2025.

Kohsei Matsutani, Shota Takashiro, Gouki Minegishi, Takeshi Kojima, Yusuke Iwasawa, and Yutaka Matsuo. RL squeezes, SFT expands: A comparative study of reasoning LLMs. In *The Fourteenth International Conference on Learning Representations*, 2026.

Jincheng Mei, Wesley Chung, Valentin Thomas, Bo Dai, Csaba Szepesvari, and Dale Schuurmans. The role of baselines in policy gradient optimization. *Advances in Neural Information Processing Systems*, 35:17818–17830, 2022.

Andriy Mnih and Danilo Rezende. Variational inference for monte carlo objectives. In *International Conference on Machine Learning*, pp. 2188–2196. PMLR, 2016.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pp. 1928–1937. PmLR, 2016.

Soichiro Nishimori and Paavo Parmas. Retry policy gradients in continuous action spaces. *arXiv preprint arXiv:2606.05888*, 2026.

Soichiro Nishimori, Paavo Parmas, Sotetsu Koyamada, Tadashi Kozuno, Toshinori Kitamura, Shin Ishii, and Yutaka Matsuo. Emergence of exploration in policy gradient reinforcement learning via retrying. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=NpvBAOc87E.

Ku Onoda, Paavo Parmas, Hiroki Furuta, Soichiro Nishimori, Yuta Oshima, Shohei Taniguchi, and Yutaka Matsuo. Multi-Axis Max@K Reinforcement Learning for Representative Diversity in Text-to-Image Generation. *arXiv preprint arXiv:2607.14962*, 2026. doi: 10.48550/arXiv.2607.14962. URL https://arxiv.org/abs/2607.14962.

OpenAI, 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.

Paavo Parmas. Total stochastic gradient algorithms and applications in reinforcement learning. *Advances in Neural Information Processing Systems*, 31, 2018.

Paavo Parmas and Masashi Sugiyama. A unified view of likelihood ratio and reparameterization gradients. In *International Conference on Artificial Intelligence and Statistics*, pp. 4078–4086. PMLR, 2021.

Paavo Parmas, Carl Edward Rasmussen, Jan Peters, and Kenji Doya. PIPPS: Flexible model-based policy search robust to the curse of chaos. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4065–4074. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/parmas18a.html.

Paavo Parmas, Yongmin Kim, Kohsei Matsutani, Shota Takashiro, Soichiro Nishimori, Takeshi Kojima, Yusuke Iwasawa, and Yutaka Matsuo. OrderGrad: Optimizing Beyond the Mean with Order-Statistic Policy Gradient Estimation. *arXiv preprint arXiv:2606.06096*, 2026. doi: 10.48550/ARXIV.2606.06096. URL https://arxiv.org/abs/2606.06096.

Ruotian Peng, Yi Ren, Zhouliang Yu, Weiyang Liu, and Yandong Wen. SimKO: Simple Pass@K policy optimization. *arXiv preprint arXiv:2510.14807*, 2025.

Jan Peters and Stefan Schaal. Reinforcement learning of motor skills with policy gradients. *Neural networks*, 21(4):682–697, 2008.

Shrinivas Ramasubramanian, Daman Arora, Fahim Tajwar, Guanning Zeng, Qingyang Wu, Zhongzhu Zhou, Chenfeng Xu, Haiwen Feng, Yuda Song, Aarti Singh, Ruslan Salakhutdinov, J. Andrew Bagnell, Jeff Schneider, and Andrea Zanette. Tail-Likelihood Reinforcement Learning. *arXiv preprint arXiv:2609.02987*, 2026. doi: 10.48550/ARXIV.2609.02987. URL https://arxiv.org/abs/2609.02987.

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

Amrith Setlur, Matthew YR Yang, Charlie Snell, Jeremy Greer, Ian Wu, Virginia Smith, Max Simchowitz, and Aviral Kumar. e3: Learning to explore enables extrapolation of test-time compute for LLMs. *arXiv preprint arXiv:2506.09026*, 2025.

Darsh J Shah, Peter Rushton, Somanshu Singla, Mohit Parmar, Kurt Smith, Yash Vanjani, Ashish Vaswani, Adarsh Chaluvaraju, Andrew Hojel, Andrew Ma, Anil Thomas, Anthony Polloreno, Ashish Tanwer, Burhan Drak Sibai, Divya S Mansingka, Divya Shivaprasad, Ishaan Shah, Karl Stratos, Khoi Nguyen, Michael Callahan, Michael Pust, Mrinal Iyer, Philip Monk, Platon Mazarakis, Ritvik Kapila, Saurabh Srivastava, and Tim Romanski. Rethinking reflection in pre-Training. *arXiv preprint arXiv:2504.04022*, 2025.

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*, 2024.

Han Shen. On entropy control in LLM-RL algorithms. In *The Fourteenth International Conference on Learning Representations*, 2026.

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.

Yuda Song, Julia Kempe, and Remi Munos. Outcome-based exploration for LLM reasoning. *arXiv preprint arXiv:2509.06941*, 2025a.

Yuda Song, Hanlin Zhang, Carson Eisenach, Sham M. Kakade, Dean Foster, and Udaya Ghai. Mind the gap: Examining the self-improvement capabilities of large language models. In *The Thirteenth International Conference on Learning Representations*, 2025b.

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. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2602.02710.

Yunhao Tang, Kunhao Zheng, Gabriel Synnaeve, and Remi Munos. Optimizing language models for inference time objectives using reinforcement learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 59066–59085. PMLR, 2025.

Christos Thrampoulidis, Sadegh Mahdavi, and Wenlong Deng. Advantage Shaping as Surrogate Reward Maximization: Unifying Pass@K Policy Gradients. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=R1RhBFUk8t.

Bingkui Tong, Junpei Komiyama, Soichiro Nishimori, and Paavo Parmas. Finite-time regret analysis of retry-aware bandits. *arXiv preprint arXiv:2605.20854*, 2026.

George Tucker, Andriy Mnih, Chris J Maddison, John Lawson, and Jascha Sohl-Dickstein. Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models. *Advances in Neural Information Processing Systems*, 30, 2017.

Jens Tuyls, Dylan J Foster, Akshay Krishnamurthy, and Jordan T Ash. Representation-based exploration for language models: From test-time to post-training. *arXiv preprint arXiv:2510.11686*, 2025.

Christian Walder and Deep Tejas Karkhanis. Pass@K policy optimization: Solving harder reinforcement learning problems. *Advances in Neural Information Processing Systems*, 38:152416–152445, 2025.

Zengzhi Wang, Fan Zhou, Xuefeng Li, and Pengfei Liu. OctoThinker: Mid-training incentivizes reinforcement learning scaling. *arXiv preprint arXiv:2506.20512*, 2025.

Lex Weaver and Nigel Tao. The optimal reward baseline for gradient-based reinforcement learning. *arXiv preprint arXiv:1301.2315*, 2013.

Xumeng Wen, Zihan Liu, Shun Zheng, Zhijian Xu, Shengyu Ye, Zhirong Wu, Xiao Liang, Yang Wang, Junjie Li, Ziming Miao, Jiang Bian, and Mao Yang. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base LLMs. *arXiv preprint arXiv:2506.14245*, 2025.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256, May 1992.

Cathy Wu, Aravind Rajeswaran, Yan Duan, Vikash Kumar, Alexandre M Bayen, Sham Kakade, Igor Mordatch, and Pieter Abbeel. Variance reduction for policy gradient with action-dependent factorized baselines. *arXiv preprint arXiv:1803.07246*, 2018.

Fang Wu, Weihao Xuan, Ximing Lu, Mingjie Liu, Yi Dong, Zaid Harchaoui, and Yejin Choi. The invisible leash: Why RLVR may or may not escape its origin. *arXiv preprint arXiv:2507.14843*, 2025.

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

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

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025b.

Zhaoning Yu, Will Su, Leitian Tao, Haozhu Wang, Aashu Singh, Hanchao Yu, Jianyu Wang, Hongyang Gao, Weizhe Yuan, Jason Weston, Ping Yu, and Jing Xu. RESTRAIN: From spurious votes to signals – self-driven rl with self-penalization. *arXiv preprint arXiv:2510.02172*, 2025.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Huaye Zeng, Dongfu Jiang, Haozhe Wang, Ping Nie, Xiaotong Chen, and Wenhu Chen. AceCoder: Acing coder RL via automated test-case synthesis. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2025.

Charlie Zhang, Graham Neubig, and Xiang Yue. On the interplay of pre-training, mid-training, and rl on reasoning language models. *arXiv preprint arXiv:2512.07783*, 2025a.

Kaichen Zhang, Shenghao Gao, Yuzhong Hong, Haipeng Sun, Junwei Bao, Hongfei Jiang, Yang Song, Dingqian Hong, and Hui Xiong. RSPO: Risk-Seeking Policy Optimization for Pass@k and Max@k Metrics in Large Language Models. *arXiv preprint arXiv:2508.01174*, 2025b. doi: 10.48550/ARXIV.2508.01174. URL https://arxiv.org/abs/2508.01174.

Shenao Zhang, Donghan Yu, Yihao Feng, Bowen Jin, Zhaoran Wang, John Peebles, and Zirui Wang. Learning to reason as action abstractions with scalable mid-training rl. *arXiv preprint arXiv:2509.25810*, 2025c.

Rosie Zhao, Alexandru Meterez, Sham Kakade, Cengiz Pehlevan, Samy Jelassi, and Eran Malach. Echo chamber: Rl post-training amplifies behaviors learned in pretraining. In *Second Conference on Language Modeling*, 2025.

Haizhong Zheng, Yang Zhou, Brian R Bartoldson, Bhavya Kailkhura, Fan Lai, Jiawei Zhao, and Beidi Chen. Act only when it pays: Efficient reinforcement learning for LLM reasoning via selective rollouts. *arXiv preprint arXiv:2506.02177*, 2025a.

Tianyu Zheng, Tianshun Xing, Qingshui Gu, Taoran Liang, Xingwei Qu, Xin Zhou, Yizhi Li, Zhoufutu Wen, Chenghua Lin, Wenhao Huang, et al. First return, entropy-eliciting explore. *arXiv preprint arXiv:2507.07017*, 2025b.

Yujun Zhou, Zhenwen Liang, Haolin Liu, Wenhao Yu, Kishan Panaganti, Linfeng Song, Dian Yu, Xiangliang Zhang, Haitao Mi, and Dong Yu. Evolving language models without labels: Majority drives selection, novelty promotes variation. *arXiv preprint arXiv:2509.15194*, 2025.

Terry Yue Zhuo et al. BigCodeBench: Benchmarking code generation with diverse function calls and complex instructions. In *International Conference on Learning Representations*, 2025.

A Additional Related Work

A.1 RLVR in LLMs

Reinforcement learning with verifiable rewards (RLVR) (Lambert et al., 2025; Guo et al., 2025) is a training paradigm that optimizes LLM policies in domains such as math and code using deterministic feedback from objective verifiers to ground model reasoning in provable correctness rather than subjective human preference (Christiano et al., 2017). Despite this promise (Guo et al., 2025; OpenAI et al., 2024), recent studies suggest that RLVR in LLMs primarily serves to amplify behaviors that are already present in the base model (Liu et al., 2025; Zhao et al., 2025; Shah et al., 2025). Yue et al. (2025) investigated the pass@K metric (Chen et al., 2021; Song et al., 2025b; Dang et al., 2025; Wen et al., 2025; Wu et al., 2025), which measures the probability that at least one correct solution is obtained when drawing K independent samples (i.e., best-of-K), and found experimentally that as K increases, the base model's pass@K eventually exceeds that of the RLVR-trained model. This phenomenon has been linked to diversity collapse (Dang et al., 2025; Cui et al., 2025) and squeezing reasoning paths (Matsutani et al., 2026; Hu et al., 2025; Bu et al., 2025). Several studies (Wang et al., 2025; Zhang et al., 2025c; Chen et al., 2025a; Zhang et al., 2025a) argue that mid-training is important. Consistent with this view, empirical results suggest that RLVR underperforms on Llama models (Grattafiori et al., 2024) relative to Qwen models (Yang et al., 2025b;a).

A.2 Exploration in RL for LLMs

In light of these limitations, prior studies have incorporated exploration into RLVR. Cui et al. (2025); Cheng et al. (2026); Zheng et al. (2025b); Shen (2026); Jiang et al. (2025) leveraged entropy bonuses to encourage exploration via policy uncertainty. Song et al. (2025a) proposed outcome-based exploration using UCB-style bonuses (Auer et al., 2002). Yu et al. (2025) applied self-penalization by assigning negative rewards to high-confidence answers that deviate from the majority consensus. He et al. (2025) improved exploration by up-weighting low-probability but correct trajectories, and Gao et al. (2025) adopted Random Network Distillation (RND) (Burda et al., 2018) to provide bonuses for unknown trajectories. Zhou et al. (2025) augmented training with a semantic novelty score computed from embeddings, Li et al. (2025) employed a semantic diversity score with an external semantic comparator, and Tuyls et al. (2025) computed a representation-based novelty score from hidden states to boost exploration. Liang et al. (2025) leveraged reward-model gradients to improve temperature sampling. Setlur et al. (2025) promoted in-context exploration via skill asymmetries and negative gradients, enabling reliable extrapolation with increased test-time compute.

Other studies directly optimize the pass@K metric; these are discussed in App. A.4.

A.3 Policy Gradient Estimator and Baseline

Policy gradients are a central approach for optimizing policies in RL (Williams, 1992; Greensmith et al., 2004; Parmas, 2018; Parmas & Sugiyama, 2021). The high variance of gradient estimates is a fundamental challenge in policy gradient methods, often hindering stable convergence (Williams, 1992). To mitigate this, the method of control variates—subtracting a baseline $b(x)$ from the return—is the standard variance reduction technique. Theoretically, the optimal baseline depends on the norm of the score function and the specific reward structure (Peters & Schaal, 2008; Weaver & Tao, 2013). Greensmith et al. (2004) provided a comprehensive analysis, demonstrating that while the exact optimal baseline is computationally expensive, a baseline that approximates the expected return—thereby making the advantage approximately zero-mean—captures the majority of the variance reduction benefits. This insight justifies the widespread adoption of value function baselines in modern algorithms like PPO (Schulman et al., 2017) and A3C (Mnih et al., 2016).

In domains involving discrete latent variables or sequence generation, where learning a separate value function is often unstable or costly, *sample-based baselines* have become the dominant approach. This concept was further refined in the context of variational inference by Mnih & Rezende (2016) (VIMCO) and Gu et al. (2015), which utilize the average reward of other samples in the batch (Leave-One-Out; L1O) to construct a low-variance gradient estimator for discrete variables. Specifically, Tucker et al. (2017) highlighted that such control variates are essential for training effective estimators in high-dimensional discrete spaces.

Recently, these multi-sample baseline techniques have been adapted for reasoning tasks in LLMs. Group Relative Policy Optimization (GRPO) (Shao et al., 2024) applies group-based normalization, effectively an L1O baseline, to stabilize training without a critic network. However, these standard L1O techniques rely on the linearity of the expectation operator. Wu et al. (2018) discussed the complexities of action-dependent baselines, but the specific challenges of the *max@K* objective remain underexplored. In the max@K setting, the non-linear dependency between samples introduced by the max operator renders standard L1O baselines biased, necessitating the development of our strictly unbiased Leave-Two-Out (L2O) approach.

A.4 Pass@K and Set-Level Policy Optimization

The repeated-sampling evaluation of LLMs has motivated inference-aware fine-tuning, including optimization for verifier-based best-of-N selection (Chow et al., 2025). Several methods directly optimize pass@K for binary rewards (Tang et al., 2025; Walder & Karkhanis, 2025; Peng et al., 2025; Bagirov et al., 2025; Chen et al., 2025b). RSPO (Zhang et al., 2025b) derives pass@K/max@K gradients from the probability that a response attains the maximum reward. For general scalar rewards, Walder & Karkhanis (2025) derive max@K estimators using batches larger than K. Their expected-improvement (EI) estimator is policy-gradient unbiased but has a non-centered advantage. This is the centering issue addressed by our L2O baseline.

For binary rewards, Thrampoulidis et al. (2026) unify direct pass@K policy gradients and advantage shaping through surrogate reward maximization. Whereas our analysis keeps K fixed, MaxRL (Tajwar et al., 2026) expresses the maximum-likelihood gradient for binary rewards as a weighted sum of pass@k gradients over different k. It derives approximations for finite sampling budgets. TailRL (Ramasubramanian et al., 2026) extends this view to bounded scalar rewards by integrating log tail probabilities over reward thresholds. Its gradient is a harmonic mixture of best-of-k gradients, and its finite-rollout estimators target truncated objectives.

ReMax maximizes the expected best return over repeated trials under a distribution of reward functions. This objective promotes stochastic policies under reward uncertainty, allowing exploration to emerge from reward maximization without explicit exploration bonuses (Koyamada et al., 2022; Nishimori et al., 2026). Koyamada et al. (2022) introduced this principle through resetting: resets generate branching trajectory trees, and REINFORCE-style policy gradients optimize an objective defined on the entire tree. Nishimori et al. (2026) subsequently developed the repeated-trial formulation and introduced the EI policy-gradient estimator that motivates our analysis. Follow-up works analyzed the landscape of its policy gradients in continuous action spaces (Nishimori & Parmas, 2026) and extended it to stochastic bandits, proving sublinear regret (Tong et al., 2026). With max aggregation, ROSA (GX-Chen et al., 2026) is ReMax applied to generative modeling: both optimize the expected maximum reward over a sampled action set, averaged over a distribution of reward functions. The combination of reward uncertainty and max aggregation is precisely the ReMax objective. Extending this objective to other nonlinear set functions requires only replacing the max with another aggregator; the same likelihood-ratio policy-gradient derivation and baseline argument apply unchanged. VPO (Bahlous-Boldi et al., 2026) evaluates the best candidate under random scalarizations of vector rewards (which is a special case of ReMax). Its candidate set is generated in a single autoregressive rollout, whereas our estimator uses independently sampled responses. Developed concurrently with VPO and ROSA, multi-axis max@K (Onoda et al., 2026) builds on our work by applying our EI+L2O estimator separately to reward axes to promote diversity in text-to-image generation. Its objective sums the per-axis maxima across an image set, encouraging different images to represent different target modes. Thus, multi-axis max@K, VPO, and the max variant of ROSA share the ReMax principle of evaluating the best sample under each reward function before averaging over reward functions; they differ in the reward distributions and sampling procedures used. Furthermore, SetPO (Li et al., 2026) constructed a leave-one-out-based advantage for a set-level diversity objective based on semantic similarity for LLM reasoning.

Subsequent work, OrderGrad (Parmas et al., 2026), extends our max@K estimator to weighted order-statistic objectives, including top-m averages and lower-tail criteria. Its max@K case recovers the include-one minus leave-one-out advantage $u_i - v_i$, which we derive from EI using L2O, preserving policy-gradient unbiasedness and exactly centering batch advantages for $K < B$. Our baseline view relates estimators through unbiasedness and centering (Sec. 4).

B CODE TO COMPUTE THE ESTIMATORS

Below, we provide code to compute the statistics proposed in Sec. 3, namely the EI estimator s_i and the L2O baseline b_{-i}^{L2O}.

```python
import jax.numpy as jnp
from scipy.special import gammaln

def comb(n, k):
    """Computes binomial coefficient C(n, k) in log-space."""
    return jnp.exp(gammaln(n + 1) - gammaln(k + 1) - gammaln(n - k + 1))

def compute_batch_ei(returns: jnp.ndarray, K: int) -> jnp.ndarray:
    """
    Computes the unbiased EI estimator s_i using matrix operations.
        Corresponds to Theorem E.1.
    """
    B = returns.shape[0]
    # Sort rewards: r_{(1)} <= ... <= r_{(B)}
    order = jnp.argsort(returns)
    r_sorted = returns[order]

    # Compute Difference Matrix D_{i,j} = (r_{(i)} - r_{(j)})_+
    D = jnp.maximum(r_sorted[:, None] - r_sorted[None, :], 0.0)

        # Compute Weight Vector w_j (Theorem E.1)
    # w_j = C(j-1, K-2) / C(B-1, K-1)
    ranks = jnp.arange(B)
    w = comb(ranks, K - 2) / comb(B - 1, K - 1)
    w = jnp.nan_to_num(w) # Handle cases where rank < K-2
    # s = D @ w
    s_sorted = D @ w
    # Restore original order
    return s_sorted[jnp.argsort(order)]

def compute_l2o_baseline(returns: jnp.ndarray, K: int) -> jnp.ndarray:
    """
        Computes the L2O baseline b_{-i}^{L2O} (Theorem E.2) in O(B^2).
    """
    B = returns.shape[0]
    order = jnp.argsort(returns)
    r_sorted = returns[order]

    # Difference Matrix D and Column Sums S_l
    D = jnp.maximum(r_sorted[:, None] - r_sorted[None, :], 0.0)
    S = jnp.sum(D, axis=0) # S_l = sum_k D_{k,l}

        # LOO Column Mean Matrix M_{i,l} (Theorem E.2)
    # M_{i,l} = (S_l - D_{i,l}) / (B - 1)
    M = (S[None, :] - D) / (B - 1)

    # L2O Weight Matrix W^{L2O}_{i,l}
    # Rank adjustment: (l-1) - 1 if l > i else (l-1)
    i_idx = jnp.arange(B)[:, None]
    l_idx = jnp.arange(B)[None, :]
    adjusted_rank = l_idx - (l_idx > i_idx).astype(jnp.float32)

    W = comb(adjusted_rank, K - 2) / comb(B - 2, K - 1)
    W = jnp.nan_to_num(W)
    W = W * (1.0 - jnp.eye(B)) # Set diagonal (l=i) to 0

    # b_{-i}^{L2O} = sum_l M_{i,l} * W_{i,l}
    b_sorted = jnp.sum(M * W, axis=1)
    return b_sorted[jnp.argsort(order)]

```

Listing 1: Code to compute the statistics proposed in Sec. 3

Algorithm 2 Max@K Policy Optimization

Input: Policy π_θ, objective size K, the number of questions per batch M, group size G with $2 \le K \le G-1$.
while not converged **do**
  Sample M questions $\{x_m\}_{m=1}^{M}$.
  for each question x_m **do**
    Sample G outputs $\{a_j\}_{j=1}^{G} \sim \pi_\theta(\cdot \mid x_m)$ and evaluate rewards $\{r_j\}_{j=1}^{G}$.
    Compute EI scores $\mathbf{s}$ (Theorem E.1) and the L2O baseline $\mathbf{b}^{\mathrm{L2O}}$ (Theorem E.2).
    Compute advantages: $\mathbf{Adv} = \mathbf{s} - \mathbf{b}^{\mathrm{L2O}}$.
  end for
  Update θ by maximizing Eq. (9) with $\mathbf{Adv}$.
end while

C Details of Max@K Policy Optimization

We integrate our method into modern RL algorithms for LLM post-training. Algorithm 2 summarizes the overall procedure when combining EI + L2O with group-based RL, which we call **Max@K** **P**olicy **O**ptimization (**MaxPO**). In Sec. 3.1 and Appendix E, B denotes the number of samples used to compute EI and L2O for a single input. In LLM training, this computation is applied independently within each question-specific group. Concretely, for each question x_m, we sample G outputs $\{a_j^m\}_{j=1}^{G}$ from $\pi_\theta(\cdot \mid x_m)$ and evaluate their sequence-level rewards $\{r_j^m\}_{j=1}^{G}$. Within such a group, G plays the role of B from Sec. 3.1 and Appendix E. From these rewards, we compute group-wise EI scores $\mathbf{s}^m$ and L2O baselines $\mathbf{b}^{m,\mathrm{L2O}}$. Accordingly, for each question-specific group, the estimator $\frac{K}{G}\sum_{j=1}^{G} \nabla_\theta \log \pi_\theta(a_j^m \mid x_m)\big(s_j^m - b_{-j}^{m,\mathrm{L2O}}\big)$ is unbiased as a PG estimator. Thus, our method acts as a reward transformation from raw sequence rewards to centered sequence-level advantages $(s_j^m - b_{-j}^{m,\mathrm{L2O}})_{j=1}^{G}$, which optimize the max@K objective (Eq. (2)).

While applicable to any PG-based algorithm, we instantiate our method using group-based RL algorithms such as Group Relative Policy Optimization (GRPO) (Shao et al., 2024) and Dr.GRPO (Liu et al., 2025) for their success in reasoning tasks. An optimization batch contains M questions, yielding question-specific groups $(x_m, a_{1:G}^m, r_{1:G}^m)_{m=1}^{M}$. Each output text consists of a sequence of tokens. Because rewards in reasoning tasks are typically assigned at the sequence level, we first compute a sequence-level advantage $A_j^m = s_j^m - b_{-j}^{m,\mathrm{L2O}}$ for each output text. We then broadcast this same scalar to all token positions when forming the token-level loss, i.e., $A_{j,i}^m = A_j^m$ for every token position i in output a_j^m. In practice, this corresponds to assigning the same reward to all tokens in an output text, $r_{j,i}^m = r_j^m$. Given a reference policy π_{ref}, we optimize the following loss:

$$L(\theta) = \frac{1}{M}\sum_{m=1}^{M}\frac{K}{G}\sum_{j=1}^{G}\frac{1}{|a_j^m|}\sum_{i=1}^{|a_j^m|} \min\left[\phi_\theta(x_m, a_{j,i}^m)A_{j,i}^m, \phi_\theta^\varepsilon(x_m, a_{j,i}^m)A_{j,i}^m\right] - \beta \mathrm{KL}(\pi_\theta|\pi_{\text{ref}}), \tag{9}$$

where $\beta \ge 0$ and $\varepsilon \ge 0$ are hyperparameters, and KL denotes the Kullback–Leibler (KL) divergence. We define $\phi_\theta(x, a_i) = \pi_\theta(a_i \mid x, a_{<i})/\pi_{\text{ref}}(a_i \mid x, a_{<i})$ as the ratio between our policy and the reference policy, and $\phi_\theta^\varepsilon(x, a_i)$ as its clipped version in $[1-\varepsilon, 1+\varepsilon]$. $|a_j^m|$ is the sequence length of the j-th text in the m-th group. Eq. (9) is identical to GRPO (Shao et al., 2024) except for how the advantage is computed. While standard GRPO normalizes rewards by dividing by the inner-group standard deviation, a practice that introduces bias into the PG estimator, both our method and Pass@K Policy Optimization (PKPO) (Walder & Karkhanis, 2025) avoid this division to preserve unbiasedness. The key distinction from Walder & Karkhanis (2025) lies in the advantage construction: while their approach is equivalent to using our raw EI scores, we further subtract the L2O baseline to achieve lower variance.

D PROOFS

D.1 PROOF OF PROPOSITION 3.1

In this appendix, we prove both claims in Proposition 3.1: unbiasedness of the L2O advantage in expectation and exact centering of its realized batch mean.

Proof. We assume $2 \leq K \leq B-1$ so that all subsets used below are well-defined. Let $\mathcal{D} = (a_{1:B}, r_{1:B})$ be a set of samples where each action is drawn i.i.d. from π_θ and $r_i := r(a_i)$. We define the *population Expected Improvement* μ_{EI} as the expected gain of a single action against the maximum of $K-1$ other independent samples. Formally, let $a_1, \ldots, a_K \overset{\text{i.i.d.}}{\sim} \pi_\theta$ be K independent random variables. Due to the i.i.d. assumption on $a_1, \ldots, a_K$, the population EI is defined as:

$$\mu_{EI} := \mathbb{E}_a\left[s(a)\right] = \mathbb{E}_{a_{1:K}}\left[\left(r(a_1) - \max_{k=2,\ldots,K} r(a_k)\right)_+\right]. \tag{10}$$

Unbiasedness of the EI Estimator s_i. First, we verify that s_i targets μ_{EI} following the U-statistics theory (Hoeffding, 1992). Given a batch $\mathcal{D} = (a_{1:B}, r_{1:B})$, s_i is constructed by averaging over all possible subsets of size $K-1$ from the batch excluding i (denoted as $\mathcal{U}_{-i} = \{1, \ldots, B\} \setminus \{i\}$).

$$s_i := \mathbb{E}_{\mathcal{I}}\left[\left(r_i - \max_{k\in\mathcal{I}} r_k\right)_+\right] = \frac{1}{\binom{B-1}{K-1}} \sum_{\mathcal{I}\subseteq\mathcal{U}_{-i}, |\mathcal{I}|=K-1} \left(r_i - \max_{k\in\mathcal{I}} r_k\right)_+. \tag{11}$$

Taking the expectation over random batches $\mathcal{D} \sim \pi_\theta$:

$$\mathbb{E}_{\mathcal{D}\sim\pi_\theta}\left[s_i\right] = \frac{1}{\binom{B-1}{K-1}} \sum_{\mathcal{I}\subseteq\mathcal{U}_{-i}, |\mathcal{I}|=K-1} \mathbb{E}_{\mathcal{D}\sim\pi_\theta}\left[\left(r_i - \max_{k\in\mathcal{I}} r_k\right)_+\right]. \tag{12}$$

For any fixed subset $\mathcal{I}$ of size $K-1$, the set of indices $\{i\} \cup \mathcal{I}$ constitutes K distinct samples. Again, by the i.i.d. assumption, the joint distribution of $\{a_i\} \cup \{a_k\}_{k\in\mathcal{I}}$ depends only on the number of samples, not their indices. Thus:

$$\mathbb{E}_{\mathcal{D}\sim\pi_\theta}\left[\left(r_i - \max_{k\in\mathcal{I}} r_k\right)_+\right] = \mathbb{E}_{a_{1:K}}\left[\left(r(a_1) - \max_{k=2,\ldots,K} r(a_k)\right)_+\right] = \mu_{EI}. \tag{13}$$

Substituting this back:

$$\mathbb{E}_{\mathcal{D}\sim\pi_\theta}\left[s_i\right] = \frac{1}{\binom{B-1}{K-1}} \sum_{\mathcal{I}\subseteq\mathcal{U}_{-i}, |\mathcal{I}|=K-1} \mu_{EI} = \mu_{EI}. \tag{14}$$

Unbiasedness of the L2O Baseline b_{-i}^{L2O}. Next, we show that the baseline also targets μ_{EI}. The L2O baseline is defined as the average of leave-two-out estimators:

$$b_{-i}^{L2O} := \frac{1}{B-1} \sum_{j\neq i} s_j^{(-i)}. \tag{15}$$

Here, $s_j^{(-i)}$ is the EI estimator for sample j computed using indices $\mathcal{U}_{-ij} = \{1, \ldots, B\} \setminus \{i, j\}$. Crucially, to estimate the same quantity μ_{EI} (which involves a 1-vs-$(K-1)$ comparison), $s_j^{(-i)}$ must aggregate over subsets of size $K-1$:

$$s_j^{(-i)} := \frac{1}{\binom{B-2}{K-1}} \sum_{\mathcal{I}'\subseteq\mathcal{U}_{-ij}, |\mathcal{I}'|=K-1} \left(r_j - \max_{k\in\mathcal{I}'} r_k\right)_+. \tag{16}$$

By the linearity of expectation:

$$\mathbb{E}_{\mathcal{D}\sim\pi_\theta}\left[b_{-i}^{L2O}\right] = \frac{1}{B-1} \sum_{j\neq i} \mathbb{E}_{\mathcal{D}\sim\pi_\theta}\left[s_j^{(-i)}\right]. \tag{17}$$

Focusing on a single term $\mathbb{E}_{\mathcal{D}\sim\pi_\theta}\left[s_j^{(-i)}\right]$:

$$\mathbb{E}_{\mathcal{D}\sim\pi_\theta}\left[s_j^{(-i)}\right] = \frac{1}{\binom{B-2}{K-1}} \sum_{\mathcal{I}'\subseteq\mathcal{U}_{-ij},|\mathcal{I}'|=K-1} \mathbb{E}_{\mathcal{D}\sim\pi_\theta}\left[\left(r_j - \max_{k\in\mathcal{I}'} r_k\right)_+\right]. \tag{18}$$

Similar to the proof of the unbiasedness of the EI estimator, for any subset $\mathcal{I}'$ of size $K-1$ drawn from $\mathcal{U}_{-ij}$, the set $\{j\}\cup\mathcal{I}'$ consists of K distinct i.i.d. samples. Thus, by the i.i.d. assumption, we have:

$$\mathbb{E}_{\mathcal{D}\sim\pi_\theta}\left[\left(r_j - \max_{k\in\mathcal{I}'} r_k\right)_+\right] = \mathbb{E}_{a_{1:K}}\left[\left(r(a_1) - \max_{k=2,\dots,K} r(a_k)\right)_+\right] = \mu_{EI}. \tag{19}$$

Summing over all subsets:

$$\mathbb{E}_{\mathcal{D}\sim\pi_\theta}\left[s_j^{(-i)}\right] = \frac{1}{\binom{B-2}{K-1}} \cdot \binom{B-2}{K-1} \cdot \mu_{EI} = \mu_{EI}. \tag{20}$$

Finally, averaging over $j \neq i$:

$$\mathbb{E}_{\mathcal{D}\sim\pi_\theta}\left[b_{-i}^{L2O}\right] = \frac{1}{B-1}\sum_{j\neq i} \mu_{EI} = \mu_{EI}. \tag{21}$$

Exact Centering of the Realized Batch Mean. We now show that the batch-average L2O advantage is exactly zero for every realized batch $\mathcal{D}$. It suffices to prove that

$$\sum_{i=1}^{B} b_{-i}^{L2O} = \sum_{i=1}^{B} s_i. \tag{22}$$

Fix $j \in \{1,\dots,B\}$ and consider the average of the leave-two-out EI estimators targeting sample j over all removed indices $i \neq j$:

$$\frac{1}{B-1}\sum_{i\neq j} s_j^{(-i)} = \frac{1}{B-1}\sum_{i\neq j} \frac{1}{\binom{B-2}{K-1}} \sum_{\mathcal{I}\subseteq\{1,\dots,B\}\setminus\{i,j\},\,|\mathcal{I}|=K-1} \left(r_j - \max_{k\in\mathcal{I}} r_k\right)_+. \tag{23}$$

Re-index the inner sum by subsets $\mathcal{I}\subseteq\mathcal{U}_{-j} := \{1,\dots,B\}\setminus\{j\}$ of size $K-1$. For any fixed such subset $\mathcal{I}$, the term $(r_j - \max_{k\in\mathcal{I}} r_k)_+$ appears once for each removed index $i \notin \mathcal{I}\cup\{j\}$. There are exactly $B-K$ such choices of i. Therefore,

$$\frac{1}{B-1}\sum_{i\neq j} s_j^{(-i)} = \frac{B-K}{(B-1)\binom{B-2}{K-1}} \sum_{\mathcal{I}\subseteq\mathcal{U}_{-j},\,|\mathcal{I}|=K-1} \left(r_j - \max_{k\in\mathcal{I}} r_k\right)_+ \tag{24}$$

$$= \frac{1}{\binom{B-1}{K-1}} \sum_{\mathcal{I}\subseteq\mathcal{U}_{-j},\,|\mathcal{I}|=K-1} \left(r_j - \max_{k\in\mathcal{I}} r_k\right)_+ \tag{25}$$

$$= s_j, \tag{26}$$

where the second equality uses the combinatorial identity

$$\frac{B-K}{(B-1)\binom{B-2}{K-1}} = \frac{1}{\binom{B-1}{K-1}}. \tag{27}$$

Summing this identity over j yields

$$\sum_{i=1}^{B} b_{-i}^{L2O} = \frac{1}{B-1}\sum_{i=1}^{B}\sum_{j\neq i} s_j^{(-i)} = \sum_{j=1}^{B}\frac{1}{B-1}\sum_{i\neq j} s_j^{(-i)} = \sum_{j=1}^{B} s_j. \tag{28}$$

Hence,

$$\frac{1}{B}\sum_{i=1}^{B}\left(s_i - b_{-i}^{L2O}\right) = 0 \tag{29}$$

for every realized batch $\mathcal{D}$.

Conclusion. Combining Eq. (14), Eq. (21), and the exact batch-centering identity above, we conclude that the L2O advantage is unbiased in expectation and exactly centered at the batch level. □

E Derivation for Efficient Computation

E.1 Computation for the EI Estimator

First, we demonstrate how to efficiently compute the EI vector $\mathbf{s} = [s_{(1)}, \ldots, s_{(B)}]^\top$ using its rank structure. Assume rewards are sorted in ascending order: $r_{(1)} \leq \cdots \leq r_{(B)}$. Let $W = \max_{k \in \mathcal{I}} r_{(k)}$ be the maximum reward in the batch. For a target sample i, the term $\big(r_{(i)} - W\big)_+$ is non-zero only if $W < r_{(i)}$. The probability that the maximum W equals a specific value $r_{(j)}$ (where $j < i$) follows a hypergeometric distribution depending only on the rank j: $P(W = r_{(j)}) := \binom{j-1}{K-2} / \binom{B-1}{K-1}$, for $j < i$. Here, the denominator represents the total number of ways to choose $K-1$ samples from $B-1$ candidates. The numerator corresponds to fixing $r_{(j)}$ as the maximum and choosing the remaining $K-2$ samples from the $j-1$ candidates smaller than $r_{(j)}$. Consequently, the estimator $s_{(i)}$ can be expressed as a weighted sum of pairwise ReLU differences:

$$s_{(i)} = \sum_{j=1}^{B} P(W = r_{(j)}) \big(r_{(i)} - r_{(j)}\big)_+, \tag{30}$$

This formulation leads directly to a matrix-based computation: multiplying the ReLU-difference matrix by the probability weights.

Theorem E.1 (Efficient Vectorized Computation of Batch EI)**.** *Let $\mathbf{r} \in \mathbb{R}^B$ be the sorted reward vector. Define the ReLU-difference matrix $D \in \mathbb{R}^{B \times B}$ as $D_{i,j} = \big(r_{(i)} - r_{(j)}\big)_+$, and the weight vector $\mathbf{w} \in \mathbb{R}^B$ as $w_j = \binom{j-1}{K-2} / \binom{B-1}{K-1}$. The vector of EI estimators $\mathbf{s}$ is given by:*

$$\mathbf{s} = D\mathbf{w}. \tag{31}$$

We now prove Theorem E.1, which shows that the EI estimator can be computed efficiently using matrix operations.

Proof. Assume $2 \leq K \leq B$. Let the batch rewards be sorted as $r_{(1)} \leq \cdots \leq r_{(B)}$, and fix a rank index $i \in \{1, \ldots, B\}$. By definition,

$$s_{(i)} := \mathbb{E}_{\mathcal{I}} \left[\left(r_{(i)} - \underbrace{\max_{j \in \mathcal{I}} r_{(j)}}_{:=W} \right)_+ \right], \tag{32}$$

$$= \sum_{j=1}^{B} P(W = r_{(j)}) \left(r_{(i)} - r_{(j)}\right)_+, \tag{33}$$

where the probability is for the distribution of the uniformly sampled subset $\mathcal{I}$ of size $K-1$ from the comparator pool $\mathcal{U}_{-i} = \{1, \ldots, B\} \setminus \{i\}$. Since $(r_{(i)} - r_{(j)})_+ = 0$ for all $j \geq i$, only ranks $j < i$ can contribute to the expectation. Therefore, it suffices to compute $P(W = r_{(j)})$ for $j < i$.

Fix $j < i$. The event $\{W = r_{(j)}\}$ occurs if and only if $j \in \mathcal{I}$ and the remaining $K-2$ indices in $\mathcal{I}$ are chosen from the $j-1$ indices with rewards strictly smaller than $r_{(j)}$. The number of such subsets is $\binom{j-1}{K-2}$. The total number of comparator subsets is $\binom{B-1}{K-1}$. Hence, for $j < i$,

$$P(W = r_{(j)}) = \frac{\binom{j-1}{K-2}}{\binom{B-1}{K-1}}. \tag{34}$$

Using the law of total expectation and the fact that only $j < i$ contributes, we obtain

$$s_{(i)} = \sum_{j=1}^{B} P(W = r_{(j)}) \, (r_{(i)} - r_{(j)})_+ = \sum_{j=1}^{B} w_j \, D_{i,j}, \tag{35}$$

where $D_{i,j} := (r_{(i)} - r_{(j)})_+$ and $w_j := \binom{j-1}{K-2}/\binom{B-1}{K-1}$. Stacking the identities for all i yields $\mathbf{s} = D\mathbf{w}$. □

Complexity Analysis. Our method requires $O(B \log B)$ time for sorting and $O(B^2)$ time to construct the pairwise difference matrix and perform the resulting matrix–vector multiplication. We note that Walder & Karkhanis (2025) derived an algorithm that computes the EI estimator in $O(B \log B + K)$. In the context of LLM training, the batch size B (typically the group size, e.g., 16–64) is small. Moreover, our vectorized formulation maps directly to GPU-friendly dense primitives (e.g., broadcasted elementwise operations and matrix–vector products), making it straightforward to leverage GPU parallelism in practice.

E.2 Computation for the L2O Baseline (proof of Theorem 3.2)

As with the EI estimator, we compute the L2O baseline using matrix multiplication. To this end, we rewrite b_{-i}^{L2O} as a weighted sum of ReLU differences:

$$b_{-i}^{\mathrm{L2O}} = \frac{1}{B-1} \sum_{l \neq i}^{B} P(W_{-i} = r_{(l)}) \sum_{j \neq i} \big(r_{(j)} - r_{(l)}\big)_+, \tag{36}$$

where W_{-i} is the maximum of a subset drawn from the population excluding i. Compared with Eq. (30), this form also admits a matrix computation of the weighted sum of ReLU differences.

Theorem E.2 (Efficient Computation of L2O Baseline)**.** *Let $S_l = \sum_{k=1}^{B} D_{k,l}$ be the column sums of D. Define the Leave-One-Out column mean matrix $\mathcal{M} \in \mathbb{R}^{B \times B}$ as $\mathcal{M}_{i,l} = (S_l - D_{i,l})/(B-1)$. Then, the vector of L2O baselines $\mathbf{b}^{\mathrm{L2O}} \in \mathbb{R}^B$ is given by the row-wise dot product:*

$$b_{-i}^{\mathrm{L2O}} = \sum_{l=1}^{B} \mathcal{M}_{i,l}\, W_{i,l}^{\mathrm{L2O}}, \text{where} \quad W_{i,l}^{\mathrm{L2O}} = \begin{cases} 0 & \text{if } l = i, \\ \dfrac{\binom{(l-1)-\mathbb{I}[l>i]}{K-2}}{\binom{B-2}{K-1}} & \text{if } l \neq i. \end{cases} \tag{37}$$

Here, the indicator $\mathbb{I}[l > i]$ adjusts the rank of $r_{(l)}$ after removing index i from the sorted batch.

We now prove Theorem E.2, which shows that the L2O baseline can be computed efficiently using matrix operations.

Proof. Assume $2 \leq K \leq B-1$. Let $r_{(1)} \leq \cdots \leq r_{(B)}$ be the sorted rewards. Define $D \in \mathbb{R}^{B \times B}$ by $D_{j,l} := \big(r_{(j)} - r_{(l)}\big)_+$. Fix an index $i \in \{1, \ldots, B\}$. Recall that

$$b_{-i}^{\mathrm{L2O}} := \frac{1}{B-1} \sum_{j \neq i} s_j^{(-i)}. \tag{38}$$

For each $j \neq i$, define $s_j^{(-i)}$ by sampling $\mathcal{I}'$ uniformly without replacement among all subsets of size $K-1$ from $\{1, \ldots, B\} \setminus \{i, j\}$ and setting $W_{-i} := \max_{k \in \mathcal{I}'} r_{(k)}$. Then

$$s_j^{(-i)} = \mathbb{E}_{\mathcal{I}'}\Big[\big(r_{(j)} - W_{-i}\big)_+\Big] = \sum_{l \in \{1,\ldots,B\} \setminus \{i,j\}} P\big(W_{-i} = r_{(l)}\big)\ \big(r_{(j)} - r_{(l)}\big)_+. \tag{39}$$

Note that $D_{j,l} = (r_{(j)} - r_{(l)})_+ = 0$ for all $l \geq j$. For $l \geq j$, we have $D_{j,l} = (r_{(j)} - r_{(l)})_+ = 0$, hence these terms do not contribute regardless of the value assigned to $W_{i,l}^{\mathrm{L2O}}$. Therefore, it suffices to compute $P(W_{-i} = r_{(l)})$ for $l < j$.

Fix $l < j$ with $l \neq i$. The event $\{W_{-i} = r_{(l)}\}$ occurs if and only if $l \in \mathcal{I}'$ and the remaining $K-2$ indices in $\mathcal{I}'$ are chosen from indices with rank strictly below l. Among $\{1, \ldots, l-1\}$, the only index that may be excluded from the sampling pool $\{1, \ldots, B\} \setminus \{i, j\}$ is i when $i < l$. Since $l < j$, excluding j does not remove any index from $\{1, \ldots, l-1\}$. Thus, the number of available indices strictly below l in the pool is $(l-1) - \mathbb{I}[l > i]$. Hence, the number of valid subsets $\mathcal{I}'$ of size $K-1$

for which $W_{-i} = r_{(l)}$ is $\binom{(l-1)-\mathbb{I}[l>i]}{K-2}$. The total number of possible subsets $\mathcal{I}'$ is $\binom{B-2}{K-1}$. Therefore, for any $l < j$ with $l \neq i$,

$$P\big(W_{-i} = r_{(l)}\big) = \frac{\binom{(l-1)-\mathbb{I}[l>i]}{K-2}}{\binom{B-2}{K-1}} = W_{i,l}^{\mathrm{L2O}}. \tag{40}$$

Crucially, the right-hand side depends on i and l but not on j, as long as $l < j$.

Substituting (40) into (39) and using $D_{j,l} = (r_{(j)} - r_{(l)})_+$ yields

$$s_j^{(-i)} = \sum_{l=1}^{B} W_{i,l}^{\mathrm{L2O}}\, D_{j,l}. \tag{41}$$

Averaging over $j \neq i$ and exchanging the finite sums gives

$$b_{-i}^{\mathrm{L2O}} = \frac{1}{B-1} \sum_{j\neq i} \sum_{l=1}^{B} W_{i,l}^{\mathrm{L2O}}\, D_{j,l} = \sum_{l=1}^{B} W_{i,l}^{\mathrm{L2O}}\, \frac{1}{B-1} \sum_{j\neq i} D_{j,l}. \tag{42}$$

Let $S_l := \sum_{k=1}^{B} D_{k,l}$. Then

$$\frac{1}{B-1} \sum_{j\neq i} D_{j,l} = \frac{S_l - D_{i,l}}{B-1} =: \mathcal{M}_{i,l}. \tag{43}$$

Substituting into (42) yields

$$b_{-i}^{\mathrm{L2O}} = \sum_{l=1}^{B} \mathcal{M}_{i,l}\, W_{i,l}^{\mathrm{L2O}}. \tag{44}$$

This holds for each $i \in \{1, \ldots, B\}$, which completes the proof. □

F Unified View of Estimators

In this appendix, we provide a unified view of our proposed estimators and those of Walder & Karkhanis (2025) from the perspective of marginal statistics.

F.1 Definitions of Marginal Statistics

Let $\mathcal{B} = \{1, \ldots, B\}$ be the set of indices for the batch of samples. We assume a fixed batch size B and a subset size K such that $2 \leq K \leq B-1$. For any subset $S \subseteq \mathcal{B}$, let $M(S) := \max_{j\in S} r_j$. All expectations below are finite averages over subsets, equivalent to sampling uniformly without replacement.

We formally define the two key statistics u_i and v_i for a specific index $i \in \mathcal{B}$:

- **Conditional Expected Max** u_i: The expected maximum of a size-K group conditioned on containing i:
$$u_i := \frac{1}{\binom{B-1}{K-1}} \sum_{S\subseteq\mathcal{B}\setminus\{i\},|S|=K-1} \max(r_i, M(S)). \tag{45}$$
- **Leave-One-Out Expected Max** v_i: The expected maximum of a size-K group drawn from the pool excluding i:
$$v_i := \frac{1}{\binom{B-1}{K}} \sum_{T\subseteq\mathcal{B}\setminus\{i\},|T|=K} M(T). \tag{46}$$

The quantity $\gamma(u_i - v_i)$, where $\gamma = K/B$, represents the exact marginal contribution of sample i to the max@K objective under finite without-replacement sampling. We first record the finite-batch centering proof for this canonical signal; the equivalence between EI+L2O and this canonical signal is proved in the following subsection.

Centering of the canonical finite-batch advantage. The leave-one-out term v_i depends only on rewards in $\mathcal{B} \setminus \{i\}$, so it is a valid leave-one-out baseline for response i and preserves policy-gradient unbiasedness. It remains to prove that the realized batch average of $u_i - v_i$ is exactly zero. Summing the definitions over all i, every size-K subset $V \subseteq \mathcal{B}$ contributes to u_i exactly when $i \in V$. Since $|V| = K$, each $M(V)$ is counted exactly K times:

$$\sum_{i=1}^{B} u_i = \frac{K}{\binom{B-1}{K-1}} \sum_{\substack{V \subseteq \mathcal{B} \\ |V|=K}} M(V). \tag{47}$$

Similarly, a size-K subset V contributes to v_i exactly when $i \notin V$. There are $B - K$ such indices, so

$$\sum_{i=1}^{B} v_i = \frac{B-K}{\binom{B-1}{K}} \sum_{\substack{V \subseteq \mathcal{B} \\ |V|=K}} M(V). \tag{48}$$

The coefficients are equal because

$$\frac{K}{\binom{B-1}{K-1}} = \frac{B-K}{\binom{B-1}{K}}. \tag{49}$$

Therefore $\sum_i u_i = \sum_i v_i$, and hence

$$\frac{1}{B} \sum_{i=1}^{B} (u_i - v_i) = 0 \tag{50}$$

for every realized batch. This establishes exact centering of the canonical finite-batch advantage.

F.2 Relationship Between EI-L2O and Marginal Statistics

We now show that our bias-corrected signal $\gamma(s_i - b_i)$ is identical to the marginal contribution $\gamma(u_i - v_i)$. First, we explicitly define the finite-batch Expected Improvement s_i and the L2O baseline b_i:

- **Expected Improvement** s_i: The direct finite-sample version of $s_i = \mathbb{E}[(r_i - W)_+]$ where W is the maximum of $K - 1$ comparators:

$$s_i := \frac{1}{\binom{B-1}{K-1}} \sum_{S \subseteq \mathcal{B} \setminus \{i\}, |S|=K-1} (r_i - M(S))_+. \tag{51}$$

- **L2O Baseline** b_i: The average of leave-two-out EI values:

$$b_i := \frac{1}{B-1} \sum_{j \neq i} s_j^{(-i)}, \tag{52}$$

  where $s_j^{(-i)}$ is the EI of j computed over the pool $\mathcal{B} \setminus \{i, j\}$.

Using the identity $\max(r_i, M(S)) = M(S) + (r_i - M(S))_+$, we decompose u_i as:

$$u_i = w_i + s_i, \tag{53}$$

where w_i is the expected comparator-max when sampling $K - 1$ items from the pool excluding i. We now show that the L2O baseline removes exactly the gap between the comparator-max baseline w_i and the leave-one-out max@K baseline v_i. Fix i and expand

$$b_i = \frac{1}{B-1} \sum_{j \neq i} \frac{1}{\binom{B-2}{K-1}} \sum_{\substack{S \subseteq \mathcal{B} \setminus \{i,j\} \\ |S|=K-1}} \left(M(S \cup \{j\}) - M(S) \right). \tag{54}$$

For the first term, every size-K subset $T \subseteq \mathcal{B} \setminus \{i\}$ appears exactly K times, once for each possible choice of $j \in T$. Therefore,

$$\frac{1}{(B-1)\binom{B-2}{K-1}} \sum_{j \neq i} \sum_{\substack{S \subseteq \mathcal{B} \setminus \{i,j\} \\ |S|=K-1}} M(S \cup \{j\}) = \frac{1}{\binom{B-1}{K}} \sum_{\substack{T \subseteq \mathcal{B} \setminus \{i\} \\ |T|=K}} M(T) = v_i. \tag{55}$$

For the second term, every size-$(K-1)$ subset $S \subseteq \mathcal{B} \setminus \{i\}$ appears exactly $B-K$ times, once for each $j \in \mathcal{B} \setminus (\{i\} \cup S)$. Thus,

$$\frac{1}{(B-1)\binom{B-2}{K-1}} \sum_{j \neq i} \sum_{\substack{S \subseteq \mathcal{B} \setminus \{i,j\} \\ |S|=K-1}} M(S) = \frac{1}{\binom{B-1}{K-1}} \sum_{\substack{S \subseteq \mathcal{B} \setminus \{i\} \\ |S|=K-1}} M(S) = w_i. \tag{56}$$

Hence $b_i = v_i - w_i$. Combining this with $s_i = u_i - w_i$ yields

$$s_i - b_i = (u_i - w_i) - (v_i - w_i) = u_i - v_i. \tag{57}$$

Thus, the scaled signal $\gamma(s_i - b_i)$ exactly recovers the marginal contribution $\gamma(u_i - v_i)$.

F.3 L2O for Arbitrary Comparator-Only Baselines

The previous subsection treats EI, which corresponds to subtracting the comparator-max baseline $C(S) = M(S)$ before applying L2O. The same cancellation holds for any baseline that depends only on the comparator set.

Proposition F.1 (L2O cancels arbitrary comparator-only baselines)**.** *Fix $2 \leq K \leq B-1$ and a realized batch $\mathcal{B}$. Let $C(S)$ be any scalar-valued function of a comparator set $S \subseteq \mathcal{B}$ with $|S| = K-1$. For each $i \in \mathcal{B}$, define the comparator-baselined group signal*

$$\alpha_i^C := \frac{1}{\binom{B-1}{K-1}} \sum_{\substack{S \subseteq \mathcal{B} \setminus \{i\} \\ |S|=K-1}} \left(M(S \cup \{i\}) - C(S)\right). \tag{58}$$

Define its L2O baseline by recomputing the same signal for each $j \neq i$ after removing i:

$$b_i^C := \frac{1}{B-1} \sum_{j \neq i} \alpha_j^{C,(-i)}, \tag{59}$$

where

$$\alpha_j^{C,(-i)} := \frac{1}{\binom{B-2}{K-1}} \sum_{\substack{S \subseteq \mathcal{B} \setminus \{i,j\} \\ |S|=K-1}} \left(M(S \cup \{j\}) - C(S)\right). \tag{60}$$

Then

$$\alpha_i^C - b_i^C = u_i - v_i \tag{61}$$

for every $i \in \mathcal{B}$.

Proof. Define the averaged comparator-only baseline for response i by

$$c_i := \frac{1}{\binom{B-1}{K-1}} \sum_{\substack{S \subseteq \mathcal{B} \setminus \{i\} \\ |S|=K-1}} C(S). \tag{62}$$

By the definition of u_i, the starting signal decomposes as

$$\alpha_i^C = u_i - c_i. \tag{63}$$

It remains to compute the L2O baseline b_i^C. Expanding its definition gives

$$b_i^C = \frac{1}{B-1} \sum_{j \neq i} \frac{1}{\binom{B-2}{K-1}} \sum_{\substack{S \subseteq \mathcal{B} \setminus \{i,j\} \\ |S|=K-1}} \left(M(S \cup \{j\}) - C(S)\right). \tag{64}$$

We handle the M part and the C part separately. For the M part, every size-K subset $T \subseteq \mathcal{B} \setminus \{i\}$ appears exactly K times, once for each possible choice of $j \in T$. Therefore,

$$\frac{1}{(B-1)\binom{B-2}{K-1}} \sum_{j \neq i} \sum_{\substack{S \subseteq \mathcal{B} \setminus \{i,j\} \\ |S|=K-1}} M(S \cup \{j\}) \tag{65}$$

$$= \frac{K}{(B-1)\binom{B-2}{K-1}} \sum_{\substack{T \subseteq \mathcal{B} \setminus \{i\} \\ |T|=K}} M(T) \tag{66}$$

$$= \frac{1}{\binom{B-1}{K}} \sum_{\substack{T \subseteq \mathcal{B} \setminus \{i\} \\ |T|=K}} M(T) = v_i. \tag{67}$$

For the comparator-only baseline part, every size-$(K-1)$ subset $S \subseteq \mathcal{B} \setminus \{i\}$ appears exactly $B-K$ times, once for each choice of $j \in \mathcal{B} \setminus (\{i\} \cup S)$. Thus,

$$\frac{1}{(B-1)\binom{B-2}{K-1}} \sum_{j \neq i} \sum_{\substack{S \subseteq \mathcal{B} \setminus \{i,j\} \\ |S|=K-1}} C(S) \tag{68}$$

$$= \frac{B-K}{(B-1)\binom{B-2}{K-1}} \sum_{\substack{S \subseteq \mathcal{B} \setminus \{i\} \\ |S|=K-1}} C(S) \tag{69}$$

$$= \frac{1}{\binom{B-1}{K-1}} \sum_{\substack{S \subseteq \mathcal{B} \setminus \{i\} \\ |S|=K-1}} C(S) = c_i. \tag{70}$$

Combining the two parts yields

$$b_i^C = v_i - c_i. \tag{71}$$

Therefore,

$$\alpha_i^C - b_i^C = (u_i - c_i) - (v_i - c_i) = u_i - v_i. \tag{72}$$

□

This proposition includes EI as the special case $C(S) = M(S)$, for which $\alpha_i^C = s_i$, and the raw conditional max signal as the special case $C(S) = 0$, for which $\alpha_i^C = u_i$. The comparator-only condition is essential: if the added term depends on the target response itself, then the leave-out property and the cancellation above need not hold.

F.4 Comparison with Walder & Karkhanis (2025)'s estimators

Walder & Karkhanis (2025) provide two primary advantage estimators: `sloo_minus_one` (their Eq. 33) and `sloo` (their Eq. 29). Here, we formally show the equivalence between EI and their `sloo_minus_one` estimator (max@K - max@(K-1)). We then describe how the other `sloo` estimator is biased differently from `sloo_minus_one`.

`sloo_minus_one` and EI-only Signal

The `sloo_minus_one` variant is defined as the difference between the max of a size-K subset and the max of the same subset excluding index i.

$$s_i^{loo-1} = \frac{1}{\binom{B}{K}} \sum_{S \subseteq \mathcal{B} \setminus \{i\}, |S|=K-1} \left(\max(r_i, M(S)) - M(S)\right). \tag{73}$$

Using the fundamental identity relating the maximum to the ReLU (Positive Part) function, $\max(a, b) = b + (a - b)_+$, the term inside the summation becomes:

$$\max(r_i, M(S)) - M(S) = (r_i - M(S))_+. \tag{74}$$

Substituting this back into the estimator and multiplying by $\binom{B-1}{K-1}/\binom{B-1}{K-1}$ to align with the definition of Expected Improvement (s_i):

$$s_i^{loo-1} = \frac{\binom{B-1}{K-1}}{\binom{B}{K}} \left[\frac{1}{\binom{B-1}{K-1}} \sum_{S \subseteq \mathcal{B}\setminus\{i\}, |S|=K-1} (r_i - M(S))_+ \right] \tag{75}$$

$$= \frac{K}{B} s_i = \gamma s_i. \tag{76}$$

This proves that their `sloo_minus_one` is exactly equivalent to our EI-only signal scaled by γ.

SLOO AND RESCALED MARGINAL SIGNAL

Walder & Karkhanis (2025)'s `sloo` variant is defined as:

$$s_i^{\text{loo}} = S(i, K, \mathcal{B}) - \frac{1}{B-1} \sum_{j \in \mathcal{B}\setminus\{i\}} S(j, K, \mathcal{B} \setminus \{i\}). \tag{77}$$

Walder & Karkhanis (2025) define $S(i, K, \mathcal{U})$ as a normalized sum over K-subsets containing index i:

$$S(i, K, \mathcal{U}) := \frac{1}{\binom{|\mathcal{U}|}{K}} \sum_{I \subseteq \mathcal{U}, |I|=K, i \in I} \max_{t \in I} r_t. \tag{78}$$

Based on this definition, we derive the connection to our u_i and v_i:

1. **Connection to u_i**: Every K-subset $I \subseteq \mathcal{B}$ with $i \in I$ can be written as $\{i\} \cup S$ where $|S| = K-1$. Thus:

$$S(i, K, \mathcal{B}) = \frac{\binom{B-1}{K-1}}{\binom{B}{K}} u_i = \frac{K}{B} u_i. \tag{79}$$

2. **Connection to v_i**: Consider the sum over the reduced pool $U = \mathcal{B} \setminus \{i\}$:

$$\sum_{j \in U} S(j, K, U) = \frac{1}{\binom{B-1}{K}} \sum_{I \subseteq U, |I|=K} \left(\sum_{j \in I} 1 \right) M(I) = K v_i, \tag{80}$$

since each K-subset I contains exactly K indices j. Dividing by $B-1$ yields:

$$\frac{1}{B-1} \sum_{j \neq i} S(j, K, \mathcal{B} \setminus \{i\}) = \frac{K}{B-1} v_i. \tag{81}$$

Substituting these into the definition of Eq. (29):

$$s_i^{\text{loo}} = \frac{K}{B} u_i - \frac{K}{B-1} v_i = \gamma \left(u_i - \frac{B}{B-1} v_i \right). \tag{82}$$

Comparing this to our signal $\gamma(u_i - v_i)$, we see a discrepancy in the coefficient of v_i. Our method uses $u_i - v_i$, whereas Walder & Karkhanis (2025) implicitly use $u_i - \frac{B}{B-1} v_i$. Since our EI + L2O signal $u_i - v_i$ is the unbiased advantage estimator, their s_i^{loo} is biased at finite B: it underestimates the advantage by $\frac{\gamma}{B-1} v_i$. Furthermore, Walder & Karkhanis (2025) did not evaluate `sloo` experimentally, and we likewise exclude it because there is no clear justification for using a biased estimator. In summary, their `sloo_minus_one` (max@K - max@(K-1)) estimator tends to overestimate the advantage, whereas `sloo` underestimates it at finite batch sizes. This contrast highlights the theoretical motivation for our L2O-based approach.

F.5 Comparison with the Analytical pass@K Estimator of Chen et al. (2025b)

We next relate the analytical binary pass@K estimator of Chen et al. (2025b) to the marginal statistics u_i and v_i. This comparison applies to binary rewards, $r_i \in \{0, 1\}$. Let N_{neg} be the number of negative responses in the batch. For binary rewards, the all-subsets mean $\bar{m}$ is the empirical pass@K value:

$$\bar{m} = 1 - \frac{\binom{N_{\text{neg}}}{K}}{\binom{B}{K}}. \tag{83}$$

This is the group-level mean used in the analytical derivation of Chen et al. (2025b).

We show that the raw analytical numerator of Chen et al. (2025b), before division by the group standard deviation, is exactly

$$u_i - \bar{m}. \tag{84}$$

There are two cases.

Positive response. If $r_i = 1$, then every size-K subset containing i has maximum reward one. Therefore,

$$u_i = 1. \tag{85}$$

The raw analytical numerator for a positive response is the positive group reward minus the group mean:

$$A_i^{\text{Chen,raw}} = 1 - \bar{m} = u_i - \bar{m}. \tag{86}$$

Negative response. If $r_i = 0$, then a size-K subset containing i is positive if and only if at least one of the other $K - 1$ responses is positive. Hence

$$u_i = 1 - \frac{\binom{N_{\text{neg}}-1}{K-1}}{\binom{B-1}{K-1}}. \tag{87}$$

The raw analytical numerator for a negative response is the probability that a group containing this response is positive, minus the group mean:

$$A_i^{\text{Chen,raw}} = 1 - \frac{\binom{N_{\text{neg}}-1}{K-1}}{\binom{B-1}{K-1}} - \bar{m} = u_i - \bar{m}. \tag{88}$$

Thus, for both positive and negative responses,

$$A_i^{\text{Chen,raw}} = u_i - \bar{m}. \tag{89}$$

We next relate $u_i - \bar{m}$ to the canonical signal $u_i - v_i$. Partitioning all size-K subsets of $\mathcal{B}$ into those containing i and those not, and using $\binom{B-1}{K-1}/\binom{B}{K} = K/B$ and $\binom{B-1}{K}/\binom{B}{K} = (B-K)/B$, the all-subsets mean satisfies

$$\bar{m} = \frac{K}{B}\, u_i + \frac{B-K}{B}\, v_i. \tag{90}$$

Subtracting both sides from u_i yields

$$u_i - \bar{m} = \frac{B-K}{B}\,(u_i - v_i). \tag{91}$$

Using Eq. (91), we obtain

$$A_i^{\text{Chen,raw}} = u_i - \bar{m} = \frac{B-K}{B}(u_i - v_i). \tag{92}$$

Therefore, in the binary setting, the raw analytical estimator of Chen et al. (2025b) has the same centered direction as MaxPO up to a fixed scale.

The estimator used in Chen et al. (2025b) further divides this numerator by a group standard deviation σ_{group}. Thus,

$$A_i^{\text{Chen,std}} = \frac{u_i - \bar{m}}{\sigma_{\text{group}}}. \tag{93}$$

This normalization may be useful as an optimization heuristic, but it introduces a random batch-dependent scale. Consequently, the numerator is proportional to the MaxPO marginal signal, while the standard-deviation-normalized coefficient is not the exact no-std policy-gradient estimator derived from $u_i - v_i$.

F.6 Comparison with the All-Subsets max@K Transform of Bagirov et al. (2025)

We now compare MaxPO with the on-policy all-subsets transformation of Bagirov et al. (2025). The derivation in this subsection is on-policy and does not include the probability-ratio correction used in their off-policy estimator.

Let $\tilde{r}_i$ (Eq. (9) of Bagirov et al. (2025)) denote the raw all-subsets transformed reward assigned to sample i:

$$\tilde{r}_i := \frac{1}{\binom{B}{K}} \sum_{\substack{I \subseteq \mathcal{B} \\ |I|=K,\ i \in I}} M(I). \tag{94}$$

Every size-K subset I containing i can be written as $I = S \cup \{i\}$, where $S \subseteq \mathcal{B} \setminus \{i\}$ and $|S| = K-1$. Therefore,

$$\begin{aligned} \tilde{r}_i &= \frac{1}{\binom{B}{K}} \sum_{\substack{S \subseteq \mathcal{B} \setminus \{i\} \\ |S|=K-1}} M(S \cup \{i\}) \\ &= \frac{\binom{B-1}{K-1}}{\binom{B}{K}} u_i = \frac{K}{B} u_i. \end{aligned} \tag{95}$$

Thus, the raw on-policy transformation of Bagirov et al. (2025) is an uncentered all-subsets marginal signal.

Next, consider ordinary mean-centering of the transformed rewards. Since

$$\frac{1}{B} \sum_{i=1}^{B} u_i = \bar{m}, \tag{96}$$

we have

$$\bar{\tilde{r}} := \frac{1}{B} \sum_{i=1}^{B} \tilde{r}_i = \frac{K}{B} \bar{m}. \tag{97}$$

Therefore,

$$\begin{aligned} \tilde{r}_i - \bar{\tilde{r}} &= \frac{K}{B}(u_i - \bar{m}) \\ &= \frac{K}{B} \cdot \frac{B-K}{B}(u_i - v_i) \\ &= \frac{K(B-K)}{B^2}(u_i - v_i). \end{aligned} \tag{98}$$

Thus, the mean-centered on-policy all-subsets transform of Bagirov et al. (2025) recovers the same centered direction as MaxPO up to a fixed scale.

If one instead applies z-score normalization to the transformed rewards, the numerator remains proportional to $u_i - v_i$, but the denominator introduces a random batch-dependent scale:

$$\frac{\tilde{r}_i - \bar{\tilde{r}}}{\mathrm{std}(\tilde{r})} = \frac{\frac{K(B-K)}{B^2}(u_i - v_i)}{\mathrm{std}(\tilde{r})}. \tag{99}$$

As with the standard-deviation-normalized analytical pass@K estimator, this changes the exact no-std policy-gradient coefficient.

Finally, the off-policy estimator of Bagirov et al. (2025) is not simply another baseline choice for u_i or v_i. It introduces probability ratios for samples drawn from an older policy and then applies a first-order approximation to the product of ratios. This produces a different approximate estimator. The equivalences above therefore apply only to the on-policy all-subsets transformation and its mean-centered or z-score-normalized variants.

Clarifying "BoN mean" versus mean-centered EI. A possible source of ambiguity is the phrase "BoN mean" in Bagirov et al. (2025). One possible reading is that BoN mean starts from the EI or LOO-1 signal and then applies the usual GRPO-style mean and standard-deviation normalization. Under this reading, the centered signal would be

$$A_i^{\mathrm{meanEI}} := s_i - \bar{s}, \qquad \bar{s} := \frac{1}{B}\sum_{j=1}^{B} s_j. \tag{100}$$

This is not MaxPO. Using $s_i = u_i - w_i$, we have

$$A_i^{\mathrm{meanEI}} = (u_i - w_i) - (\bar{u} - \bar{w}), \tag{101}$$

which does not simplify to $u_i - v_i$ in general. Moreover, $\bar{s}$ can depend on r_i, because r_i may appear as a comparator inside s_j for $j \neq i$. Thus, mean-centering EI is not the L2O correction and does not generally satisfy the leave-out condition required by the policy-gradient unbiasedness argument.

The all-subsets transform in Eq. (94) is a different vector. It is proportional to u_i, not to the EI signal $s_i = u_i - w_i$:

$$\tilde{r}_i = \frac{K}{B} u_i. \tag{102}$$

Therefore, centering $\tilde{r}_i$ is fundamentally different from centering s_i. As shown in Eq. (98),

$$\tilde{r}_i - \bar{\tilde{r}} = \frac{K(B-K)}{B^2}(u_i - v_i), \tag{103}$$

so the mean-centered all-subsets transform is proportional to the MaxPO direction. In short,

$$s_i - \bar{s} \not\equiv u_i - v_i, \qquad \tilde{r}_i - \bar{\tilde{r}} \propto u_i - v_i. \tag{104}$$

The distinction is entirely about which vector is being centered.

G Toy Experiments

In this section, we explain the detailed settings and additional results for the toy experiments we presented in Sec. 5.

G.1 Bandits

G.1.1 Setting

We generate random bandit instances to rigorously evaluate the statistical properties—bias and variance—of the gradient estimators. For each problem instance (seed), we sample the reward vector $\mathbf{r} \in \mathbb{R}^{|\mathcal{A}|}$ and the policy logit vector $\boldsymbol{\theta} \in \mathbb{R}^{|\mathcal{A}|}$ from a standard normal distribution:

$$\boldsymbol{\theta} \sim \mathcal{N}(0, I), \quad \mathbf{r} \sim \mathcal{N}(0, I). \tag{105}$$

The policy is defined as a softmax distribution over actions, $\pi_\theta(a) = \mathrm{softmax}(\boldsymbol{\theta})_a$. Unless otherwise stated, we fix the max@K parameter to $K = 2$ and the batch size to $B = 8$. We construct the PG estimator $\hat{g}$ using EI: s_i, EI+L2O: $s_i - b_{-i}^{\mathrm{L2O}}$, and EI+L1O: $s_i - b_{-i}^{\mathrm{L1O}}$. For example, the EI estimator for $\nabla J^K(\theta)$ takes the form $\hat{g} = \frac{K}{B}\sum_{i=1}^{B} s_i \nabla_\theta \log \pi_\theta(a_i)$.

Ground Truth and Estimators. For a fixed problem instance specified by $(\boldsymbol{\theta}, \mathbf{r})$, we compute the ground-truth max@K policy gradient $g_{\mathrm{true}} := \nabla J^K(\boldsymbol{\theta})$ analytically via the closed-form derivative of the expected improvement (Proposition 1 of Nishimori et al. (2026)). Each gradient estimator $\hat{g}$ is computed from a group of B i.i.d. actions sampled from $\pi_{\boldsymbol{\theta}}$.

Estimation Error ("Bias") Protocol. For each fixed instance $(\boldsymbol{\theta}, \mathbf{r})$, we generate N independent gradient estimates $\{\hat{g}_j\}_{j=1}^{N}$. We measure the (finite-sample) estimation error of the Monte Carlo mean relative to the ground truth:

$$\mathrm{Err}(\boldsymbol{\theta}, \mathbf{r}) := \left\| \bar{g}_N - g_{\mathrm{true}} \right\|_2, \qquad \bar{g}_N := \frac{1}{N}\sum_{j=1}^{N} \hat{g}_j. \tag{106}$$

Empirical Total Variance Protocol. To quantify estimator stability, we measure the empirical total variance, defined as the trace of the sample covariance:

$$\widehat{\mathrm{TV}}_N(\boldsymbol{\theta}, \mathbf{r}) := \mathrm{Tr}\left(\frac{1}{N}\sum_{j=1}^{N}(\hat{g}_j - \bar{g}_N)(\hat{g}_j - \bar{g}_N)^\top\right) = \frac{1}{N}\sum_{j=1}^{N}\|\hat{g}_j - \bar{g}_N\|_2^2. \tag{107}$$

Equivalently, using the second-moment identity,

$$\widehat{\mathrm{TV}}_N(\boldsymbol{\theta}, \mathbf{r}) = \frac{1}{N}\sum_{j=1}^{N}\|\hat{g}_j\|_2^2 - \|\bar{g}_N\|_2^2. \tag{108}$$

Unless stated otherwise, we use $N = 10^5$ to estimate $\widehat{\mathrm{TV}}_N$.

The total variance of score-function gradients can scale with the action-space dimensionality. To assess variance reduction in high-dimensional regimes (relevant to LLM settings), we report $\widehat{\mathrm{TV}}_N$ across varying numbers of actions.

Aggregation Across Seeds. We repeat the above procedure over $L = 100$ independent random seeds, each generating a new pair $(\boldsymbol{\theta}, \mathbf{r})$, and report the mean and standard error of $\mathrm{Err}(\boldsymbol{\theta}, \mathbf{r})$ and $\widehat{\mathrm{TV}}_N(\boldsymbol{\theta}, \mathbf{r})$ across seeds.

Sweep Configurations. For the bias plot, we vary the number of batches $N \in \{10^3, 10^4, 10^5, 10^6\}$ to assess whether the estimation error decreases with N, fixing $B = 8$ and $K = 2$. To examine variance, we vary the action space size $|\mathcal{A}| \in \{10, 50, 100, 1000\}$ (fixing $K = 2$, $B = 8$) and the comparator size $K \in \{2, 3, 4, 5, 6\}$ (fixing $B = 8$, $|\mathcal{A}| = 100$). Larger $|\mathcal{A}|$ corresponds to LLM-relevant regimes; varying K probes how the available comparator budget affects the L2O baseline.

Comparison Baselines. We compare three estimators: EI (s_i), EI+L2O ($s_i - b_{-i}^{\mathrm{L2O}}$), and EI+L1O ($s_i - b_{-i}^{\mathrm{L1O}}$), as defined in Sec. 3.

G.1.2 Action Space Size and Variance in Softmax Policy

In the bandit setting with a softmax policy, the total variance of the policy gradient estimator is inherently tied to the dimensionality of the action space, denoted by $|\mathcal{A}|$. Here, we derive this relationship analytically.

Consider the score function for a selected action a, given by $\nabla_\theta \log \pi_\theta(a) = \mathbf{e}_a - \boldsymbol{\pi}_\theta$, where $\mathbf{e}_a$ is the one-hot vector for action a and $\boldsymbol{\pi}_\theta$ is the probability vector. Assuming the gradient variance is dominated by sampling noise (i.e., neglecting the squared norm of the expected gradient, $\|\mathbb{E}[\hat{g}]\|^2 \approx 0$), we approximate the total variance by the expected squared norm of the score function:

$$V_{\text{total}}(\boldsymbol{\pi}_\theta) \approx \mathbb{E}_{a\sim\pi_\theta}\left[\|\nabla_\theta \log \pi_\theta(a)\|^2\right]. \tag{109}$$

Expanding the squared norm $\|\mathbf{e}_a - \boldsymbol{\pi}_\theta\|^2$ yields

$$\|\mathbf{e}_a - \boldsymbol{\pi}_\theta\|^2 = (\mathbf{e}_a - \boldsymbol{\pi}_\theta)^\top(\mathbf{e}_a - \boldsymbol{\pi}_\theta) \tag{110}$$

$$= \|\mathbf{e}_a\|^2 - 2\mathbf{e}_a^\top\boldsymbol{\pi}_\theta + \|\boldsymbol{\pi}_\theta\|^2 \tag{111}$$

$$= 1 - 2\pi_\theta(a) + \|\boldsymbol{\pi}_\theta\|^2. \tag{112}$$

Taking expectation over $a \sim \pi_\theta$ gives

$$V_{\text{total}}(\boldsymbol{\pi}_\theta) = \sum_{a=1}^{|\mathcal{A}|} \pi_\theta(a) \left(1 - 2\pi_\theta(a) + \|\boldsymbol{\pi}_\theta\|^2\right) \tag{113}$$

$$= \sum_{a=1}^{|\mathcal{A}|} \pi_\theta(a) - 2\sum_{a=1}^{|\mathcal{A}|} \pi_\theta(a)^2 + \|\boldsymbol{\pi}_\theta\|^2 \sum_{a=1}^{|\mathcal{A}|} \pi_\theta(a) \tag{114}$$

$$= 1 - 2\|\boldsymbol{\pi}_\theta\|^2 + \|\boldsymbol{\pi}_\theta\|^2 \tag{115}$$

$$= 1 - \|\boldsymbol{\pi}_\theta\|^2. \tag{116}$$

Eq. (116) shows that this approximation depends only on the squared ℓ_2-norm of the probability vector.

To see the effect of the action space size $|\mathcal{A}|$, consider the uniform policy (which is common at initialization), where $\pi_\theta(a) = 1/|\mathcal{A}|$ for all a. Then

$$V_{\text{total}}(\boldsymbol{\pi}_{\text{uniform}}) = 1 - \sum_{a=1}^{|\mathcal{A}|} \left(\frac{1}{|\mathcal{A}|}\right)^2 = 1 - |\mathcal{A}| \cdot \frac{1}{|\mathcal{A}|^2} = 1 - \frac{1}{|\mathcal{A}|}. \tag{117}$$

This implies that the variance increases with $|\mathcal{A}|$ and approaches 1 as $|\mathcal{A}| \to \infty$. For example, a binary bandit ($|\mathcal{A}| = 2$) yields $V_{\text{total}}(\boldsymbol{\pi}_{\text{uniform}}) = 0.5$, whereas $|\mathcal{A}| = 1000$ yields $V_{\text{total}}(\boldsymbol{\pi}_{\text{uniform}}) = 0.999$. Thus, large action spaces can exhibit substantially higher gradient variance in the early stages of learning under a softmax policy.

G.1.3 Additional Results

Here, we report additional results for the bandit setting. In particular, we plot the variance of the gradient estimator for different batch sizes $B \in \{8, 16, 32\}$.

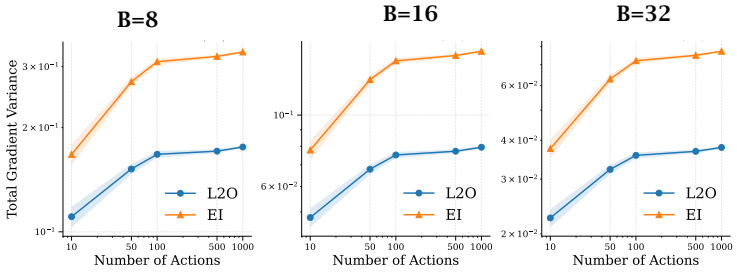


Figure 4: Variance vs action space size.

Variance vs action space size. Here, we report additional results on variance versus action space size (Figure 4). We plot the variance of the gradient estimator for action space sizes $|\mathcal{A}| \in \{10, 50, 100, 1000\}$, with batch sizes $B \in \{8, 16, 32\}$ and $K = 2$. As expected, the variance decreases as the batch size increases. Moreover, the variance-reduction effect of the L2O baseline becomes more pronounced as the action space size increases, consistent with the results in Sec. 5.1.

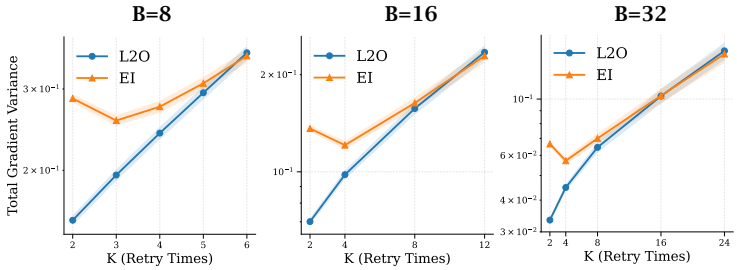


Figure 5: Variance vs K.

Variance vs K. Here, we report additional results on variance versus K (Figure 5). We plot the variance while varying K, with batch sizes $B \in \{8, 16, 32\}$ and $|\mathcal{A}| = 100$. As before, the variance decreases as the batch size increases. We also observe that the variance-reduction effect of the L2O baseline is largest when K is moderate relative to the batch size B, which is consistent with the results in Sec. 5.1.

G.2 Maze Environment

Here, we report additional results for the maze environment.

G.2.1 Setting

Environment. The maze is deterministic, with binary actions (0 or 1) available at each state. The agent receives a reward of $+1$ for each forward step along a valid path. There exists a single correct path to the goal (indicated by the red line in Fig. 6); if the agent selects the optimal action at every step, it can progress for up to 1000 steps. If the agent takes an incorrect action, the episode terminates within one or two steps. We design the maze to exhibit structural bias, where the correct action is 1 in 75% of the states. Consequently, a policy that blindly prioritizes action 1 can achieve moderately high returns, creating a local optimum that discourages exploration of the optimal path.



Figure 6: Structure of the Biased Maze Environment. The red line indicates the unique optimal path.

Policy Parameterization. We parameterize the policy using tabular logits for each state–action pair, augmented with a global bias vector, $\pi(a \mid s) = \mathrm{softmax}(\theta_{s,a} + \phi_a)$, where $\theta \in \mathbb{R}^{|S|\times|A|}$ and $\phi \in \mathbb{R}^{|A|}$. The global parameter ϕ captures an environmental bias toward specific actions and is used as a proxy for exploratory tendency, as it represents the default policy in unvisited states (since θ is initialized near 0).

Estimator Construction for Multi-Step RL. In episodic RL settings, the optimization objective is defined over the cumulative return of an episode rather than immediate rewards. Let $\tau_i = (s_{i,0}, a_{i,0}, r_{i,0}, \dots)$ denote the i-th trajectory in a batch of size B. We define the episode return as $R_i = \sum_t r_{i,t}$ and the cumulative score function as $\Psi_i = \sum_t \nabla_\theta \log \pi_\theta(a_{i,t}|s_{i,t})$. To apply the estimators derived in Section 3, we simply substitute the immediate reward r_i with the episode return R_i and the score function $\psi(x, a_i)$ with Ψ_i. The specific gradient estimators compared in our experiments are:

- **Standard REINFORCE (with L1O Baseline):** Optimizes the standard expected return $J(\theta) = \mathbb{E}_{x,a\sim\pi_\theta}[R(x, a)]$. We use the standard Leave-One-Out (L1O) baseline to reduce variance:
$$\hat{g}_{\mathrm{std}} := \frac{1}{B}\sum_{i=1}^{B} \Psi_i(R_i - b_{-i}), \quad \text{where } b_{-i} = \frac{1}{B-1}\sum_{j\neq i} R_j. \tag{118}$$
For entropy-regularized REINFORCE, we added the entropy bonus term $\beta H(\pi_\theta)$ to the objective, where $H(\pi_\theta) = -\sum_a \pi_\theta(a)\log\pi_\theta(a)$ and β is set to 0.01.
- **EI (Vanilla):** Optimizes the max@K objective using the raw Expected Improvement score s_i (calculated via Eq. (4) in Sec. 3.1 using returns $\{R_j\}_{j=1}^B$):
$$\hat{g}_{\mathrm{ei}} := \frac{K}{B}\sum_{i=1}^{B} \Psi_i s_i. \tag{119}$$
- **EI + L2O (Ours):** Optimizes the max@K objective using the centered advantage $s_i - b_{-i}^{L2O}$:
$$\hat{g}_{\mathrm{l2o}} := \frac{K}{B}\sum_{i=1}^{B} \Psi_i(s_i - b_{-i}^{L2O}), \tag{120}$$
where b_{-i}^{L2O} is the L2O baseline computed from $\{R_j\}_{j=1}^B$ as defined in Theorem E.2.

Hyperparameters. The learning rate is set to 0.03 after a grid search over $\{0.01, 0.03, 0.05\}$. We collect trajectories with batch sizes $B \in \{5, 8, 16, 32\}$ and train for 3000 iterations. We report the mean and standard error over 10 different random seeds.

Metrics. To quantify variance reduction, we measure the total variance of the gradient estimator $\mathrm{Tr}(\mathbb{V}[\hat{g}])$. In Figure 7, we report the "Grad Var Ratio", which is defined as the ratio of the gradient variance of our method (EI + L2O) to that of the vanilla EI estimator:

$$\text{Ratio} = \frac{\sum_d \text{Var}(\hat{g}_{\text{l2o}}^{(d)})}{\sum_d \text{Var}(\hat{g}_{\text{ei}}^{(d)})}, \tag{121}$$

where the sum is taken over all parameters. A ratio less than 1.0 indicates effective variance reduction.

G.2.2 Results

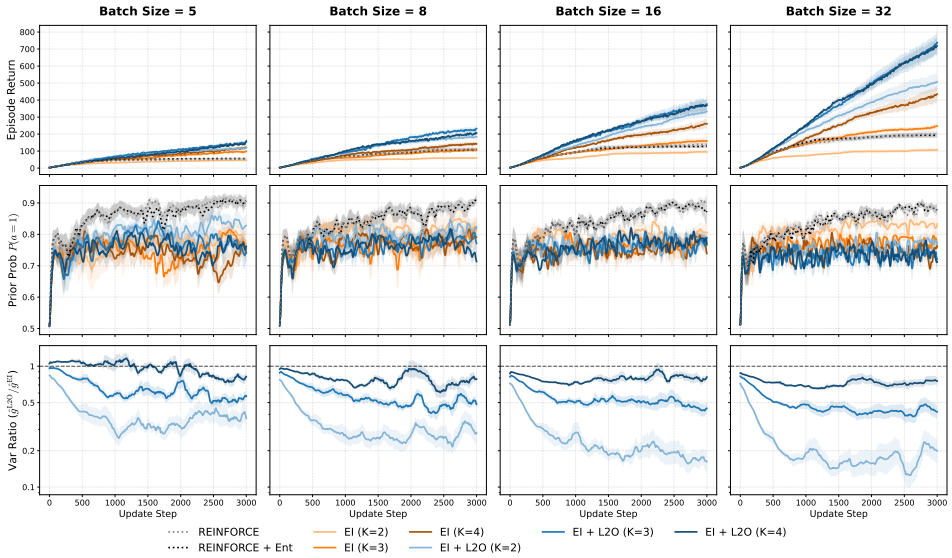


Figure 7: Average return and probability of action 1 (via global bias ϕ_a) with a moving average over 10 update steps, and gradient variance ratio with a moving average over 200 update steps, across different batch sizes $B \in \{5, 8, 16, 32\}$. The ratio uses a longer window because individual variance estimates are noisy; the two total variances are averaged before taking the ratio, and the mean and standard error over seeds are computed in log space. The error bars are computed using the standard error of the moving average. Our method (EI + L2O) consistently outperforms standard REINFORCE, Entropy-regularized REINFORCE, and vanilla EI. The probability plots indicate that EI-based methods maintain an exploratory policy (prob. ≈ 0.75) matching the environmental prior, whereas REINFORCE baselines converge to a greedy strategy (prob. > 0.9). The gradient variance ratio (EI + L2O / EI) remains below 1.0 in all settings except $B = 5$, $K = 4$, demonstrating the effectiveness of the L2O baseline.

In Figure 7, we present the full results for the maze environment across varying batch sizes $B \in \{5, 8, 16, 32\}$, reporting the average return, the probability of selecting action 1 (via the global bias parameter ϕ_a), and the gradient variance ratio. We observe that **EI + L2O consistently outperforms** all baselines—standard REINFORCE, Entropy-regularized REINFORCE, and vanilla EI—across all tested batch sizes. The probability plots reveal that EI + L2O maintains a policy with a strong exploratory tendency (action 1 probability ≈ 0.75), preventing the collapse to the suboptimal greedy strategy observed in the REINFORCE baselines. Furthermore, the gradient variance ratio is less than 1.0 throughout training in all settings except $B = 5$, $K = 4$ (discussed below), confirming that the L2O baseline effectively reduces variance. Notably, the variance-reduction effect appears more pronounced when the comparator set size is small (e.g., $K = 2$). This is expected, as increasing the number of comparators (K) tends to drive more Expected Improvement values to zero, since it

Table 2: **Task-average pass@k ($k \leq 256$).** Unweighted average over AIME24, AIME25, AMC23, MATH500, and Minerva under the evaluation protocol in Sec. 5.2.1.

Method	1	2	4	8	16	32	64	128	256
Qwen2.5-Math-7B									
GRPO	**40.7**	45.5	49.2	52.2	55.4	58.7	62.1	65.6	69.3
Entropy-Adv	38.4	43.7	48.3	52.3	56.3	60.2	64.1	67.7	70.9
PKPO	37.2	42.6	47.4	51.4	55.1	59.0	62.9	66.8	70.6
MaxPO (Ours)	39.9	**46.5**	**51.7**	**56.2**	**60.1**	**63.8**	**67.5**	**71.1**	**74.3**
Llama-3.2-3B-Instruct									
GRPO	**22.4**	**28.0**	33.3	38.2	42.9	47.7	52.7	57.5	62.1
Entropy-Adv	22.1	27.7	33.2	38.5	43.6	48.5	53.3	58.2	63.1
PKPO	20.1	25.5	31.3	37.1	42.5	47.5	52.7	58.0	63.5
MaxPO (Ours)	21.8	27.6	**33.4**	**38.7**	**43.8**	**48.9**	**54.0**	**59.5**	**65.0**

becomes harder to beat the maximum of a larger set. As a result, the additional variance reduction available from L2O naturally decreases. The only setting without a consistent reduction is $B = 5$, $K = 4$, where the ratio exceeds 1.0 early in training and stays close to 1.0 until about update step 2200 before settling around 0.8. Here, the L2O baseline must form its $K - 1 = 3$ comparators from only $B - 2 = 3$ samples; such a small comparator budget likely makes the baseline noisy, consistent with the bandit experiments.

H LLM Experiments

In this section, we provide additional results for the LLM experiments presented in Sec. 5.2.

H.1 Setting

All models are trained with a learning rate of 10^{-6}, a batch size of 1024, and an optimization mini-batch size of 256. Unless otherwise stated, we use a group size of $G = 8$ and sample each response with a temperature of 1.0. The Qwen model is trained on the MATH training split (Hendrycks et al., 2021), while the Llama model is trained on a combined dataset of the GSM8K training split (Cobbe et al., 2021) and MATH Level 1 training examples. We exclude evaluation examples from the RL training data, including all problems used in AIME24, AIME25, AMC23, MATH500, and Minerva evaluation.

We use the same learning rate for all methods to isolate the effect of the advantage estimator, rather than tuning the optimization settings separately for each method. We set the learning rate to 10^{-6} following prior RL-for-mathematical-reasoning work (Peng et al., 2025). Each Llama run is trained for 160 RL optimization steps, and each Qwen run for 140 steps. We report results from the final checkpoint of each run and perform no checkpoint selection based on evaluation-set performance.

H.2 Additional Results

Tables 2 and 3 summarize the same evaluation protocol as the main text, reporting (i) task-average pass@k up to $k \leq 256$ and (ii) per-benchmark pass@1 and pass@256. Figure 8 provides the per-benchmark pass@k curves to visualize how improvements distribute across datasets and across inference compute. In addition, Figure 9 analyzes how RL training changes the empirical support of correct solutions relative to the base model, following the taxonomy of Wu et al. (2025). Specifically, for each problem under a matched sampling budget, we categorize outcomes into *Support Preservation*, *Support Shrinkage*, *Support Expansion*, or *Out of Support* (see the caption of Figure 9).

Table 3: **Per-benchmark pass@1 / pass@256.** Results for each benchmark under the same evaluation protocol as Sec. 5.2.1.

Method	AIME24	AIME25	AMC23	MATH500	Minerva	Avg.
Qwen2.5-Math-7B						
GRPO	**30.3**/72.0	10.5/41.1	**63.3**/96.1	**76.3**/92.9	**23.0**/44.5	**40.7**/69.3
Entropy-Adv	25.6/72.3	9.8/47.3	59.5/97.0	74.7/93.6	22.4/44.2	38.4/70.9
PKPO	19.9/72.9	**11.1**/49.2	60.7/94.4	72.9/93.1	21.4/43.4	37.2/70.6
MaxPO (Ours)	28.5/**74.2**	10.6/**54.7**	62.2/**98.0**	75.6/**95.7**	22.9/**48.8**	39.9/**74.3**
Llama-3.2-3B-Instruct						
GRPO	**13.7**/50.5	0.5/34.3	**30.2**/93.5	52.0/90.5	**15.5**/41.6	**22.4**/62.1
Entropy-Adv	12.6/49.2	**0.9**/**36.5**	29.5/94.8	**52.4**/91.7	15.1/43.2	22.1/63.1
PKPO	9.3/54.4	0.5/34.0	27.4/**96.8**	49.2/90.7	14.2/41.4	20.1/63.5
MaxPO (Ours)	11.7/**58.1**	0.6/35.4	29.9/96.0	51.7/**91.8**	15.3/**43.8**	21.8/**65.0**

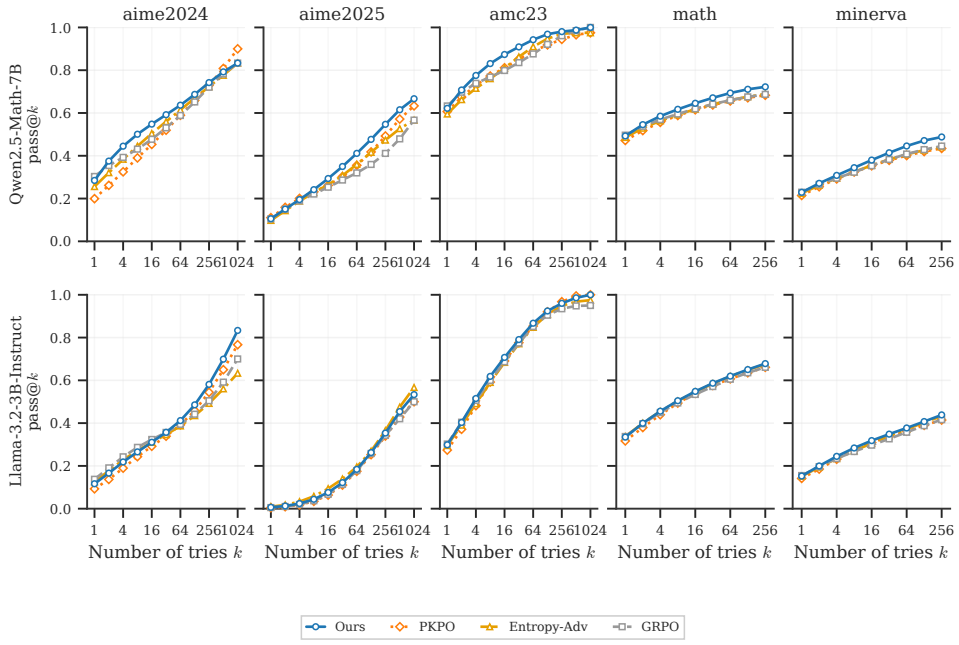


Figure 8: **Per-benchmark pass@k curves.** For each benchmark, we plot pass@k as a function of inference compute k for all methods. This complements the task-average curves in Figure 2 by showing where gains come from across datasets and inference-compute levels, and whether improvements persist as k increases.

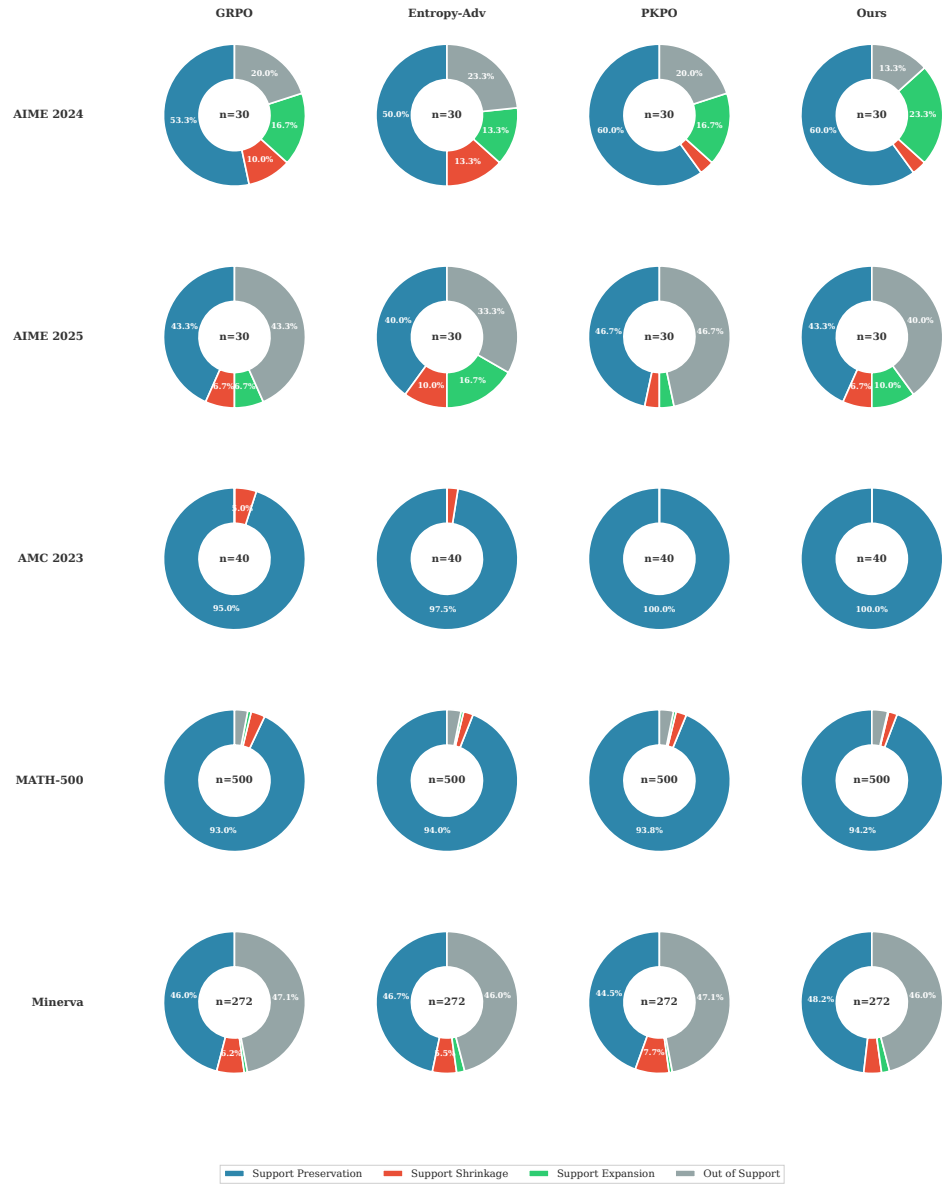


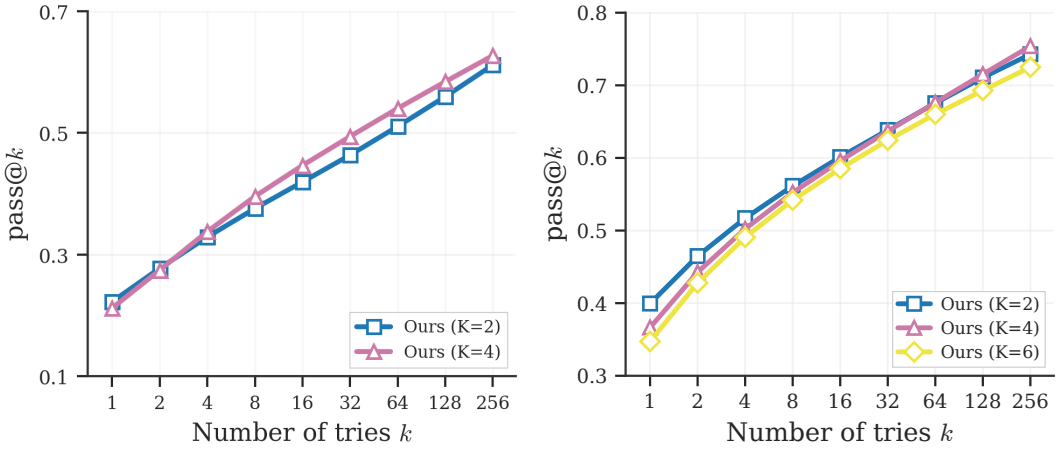
Figure 9: Empirical support dynamics relative to the base model, following the taxonomy of Wu et al. (2025). For each benchmark, we categorize each problem into one of four cases under a matched sampling budget: (i) *Support Preservation*: both the base model and the RL-trained model produce at least one correct completion; (ii) *Support Shrinkage*: the base model succeeds but the RL-trained model fails; (iii) *Support Expansion*: the RL-trained model succeeds but the base model fails; and (iv) *Out of Support*: neither model produces a correct completion. Donut charts report the proportions for each RL method across benchmarks (sample sizes are shown in the plots).

H.2.1 Effect of Objective Size K

We study the impact of the training objective size K by comparing $K \in \{2, 4\}$ while keeping other settings fixed. As shown in Figure 10, increasing K from 2 to 4 decreases pass@1 but improves pass@k for large k (e.g., $k = 256$) for both Llama and Qwen. This indicates an exploration-exploitation trade-off: increasing K promotes broader exploration, which reduces one-shot accuracy but improves the probability of success when multiple samples are available. Overall, $K = 2$ offers the best balance in our setting and is used as the default.

Table 4: **Effect of objective size K on Qwen2.5-Math-7B.** Task-average pass@k (unweighted average over AIME24, AIME25, AMC23, MATH500, and Minerva) with $G = 8$. Base denotes the model before RL training.

Method	1	2	4	8	16	32	64	128	256
Base	25.8	35.5	44.0	50.7	55.7	59.7	63.4	67.4	71.6
GRPO	**40.7**	45.5	49.2	52.2	55.4	58.7	62.1	65.6	69.3
PKPO	37.2	42.6	47.4	51.4	55.1	59.0	62.9	66.8	70.6
MaxPO ($K = 2$)	39.9	**46.5**	**51.7**	**56.2**	**60.1**	**63.8**	67.5	71.1	74.3
MaxPO ($K = 4$)	36.6	44.3	50.2	55.2	59.6	63.7	**67.6**	**71.5**	**75.4**
MaxPO ($K = 6$)	34.7	42.8	49.1	54.2	58.5	62.4	66.0	69.3	72.5



(a) Llama-3.2-3B-Instruct (b) Qwen2.5-Math-7B

Figure 10: Sensitivity to the training objective size K ($K \in \{2, 4\}$ for Llama and $K \in \{2, 4, 6\}$ for Qwen). Reported curves are task-average pass@k (unweighted average over AIME24, AIME25, AMC23, MATH500, and Minerva). For both Llama and Qwen, increasing K from 2 to 4 lowers pass@1 while boosting pass@k for large k, indicating an exploration-exploitation trade-off. On Qwen with $G = 8$, increasing K to 6 provides no additional improvement.

We further extend this ablation to $K = 6$ on Qwen2.5-Math-7B with $G = 8$ (Figure 10b and Table 4). Among the MaxPO variants, $K = 2$ performs best from pass@1 through pass@32 and is nearly tied with $K = 4$ at pass@64, whereas $K = 4$ performs slightly better at pass@128 and pass@256. $K = 6$ provides no additional improvement. We therefore use $K = 2$ as the default because it provides the best overall balance between performance at small and large evaluation values of k.

We attribute the lack of improvement at $K = 6$ mainly to a finite-group effect arising from the interaction between the objective size K and the group size G, rather than to a fundamental limitation of the estimator. Importantly, K is not simply a scale parameter: increasing K changes the optimization objective itself. In the binary-reward setting, if a group contains c correct responses, the centered advantage is non-zero only when $1 \le c \le G - K$. Thus, with $G = 8$ and $K = 6$, only groups containing one or two correct responses contribute to the update; when $c \ge 3$, every subset of six responses necessarily contains a correct response, so the empirical max@6 objective is already saturated and all centered advantages become zero. As K approaches G, the learning signal therefore becomes increasingly sparse, and increasing K alone is not expected to produce monotonic improvements. For a fixed K, increasing G widens the range $1 \le c \le G - K$, alleviates saturation, and provides a denser learning signal; larger K should therefore be paired with a sufficiently large G. This saturation is also less severe for non-binary rewards, where differences in reward magnitude can preserve non-zero advantages even when multiple responses receive positive rewards.

Table 5: **Effect of group size on Llama-3.2-3B-Instruct.** Task-average pass@k is the unweighted average over AIME24, AIME25, AMC23, MATH500, and Minerva. The final column reports the total Adam-moment variance proxy defined in Sec. 5.2.3, averaged over all training steps.

G	pass@1	pass@2	pass@4	pass@8	pass@16	pass@32	pass@64	pass@128	pass@256	Mean variance proxy
4	22.74	28.91	**35.18**	**41.17**	**46.79**	**52.21**	**57.53**	62.62	67.45	2.3873×10^{-3}
8	22.29	28.21	34.36	40.40	46.21	51.86	57.36	**62.72**	**67.88**	1.1199×10^{-3}
16	**23.05**	**29.09**	35.14	40.79	46.06	51.38	56.87	62.29	67.55	$\mathbf{6.6768\times10^{-4}}$

Table 6: **Code-generation pass@k of Qwen2.5-7B-Instruct trained with continuous rewards.** Task-average pass@k over the seven code benchmarks described in Appendix H.3. Base denotes the model before RL training.

Method	1	2	4	8	16	32	64	128	256
Base	52.6	57.4	61.5	65.0	68.0	70.5	72.5	74.2	75.6
GRPO	53.4	57.1	60.3	63.1	65.3	67.3	68.9	70.3	71.7
Entropy-Adv	53.1	56.6	59.5	62.0	64.1	65.9	67.5	68.9	70.2
PKPO	52.9	56.4	59.3	61.8	63.8	65.5	67.1	68.5	69.7
MaxPO (Ours)	**53.6**	**58.7**	**63.0**	**66.7**	**69.8**	**72.2**	**73.9**	**75.5**	**77.0**

H.2.2 Effect of Group Size G

We study the effect of group size $G \in \{4, 8, 16\}$ for MaxPO with $K = 2$ on Llama-3.2-3B-Instruct, while keeping the other training and evaluation settings fixed. As shown in Table 5, pass@k varies by at most 0.9 percentage points across the tested group sizes. In contrast, increasing G from 4 to 8 and 16 reduces the mean Adam-moment variance proxy by 53.1% and 72.0%, respectively. Thus, larger groups substantially improve gradient stability without materially changing downstream pass@k performance.

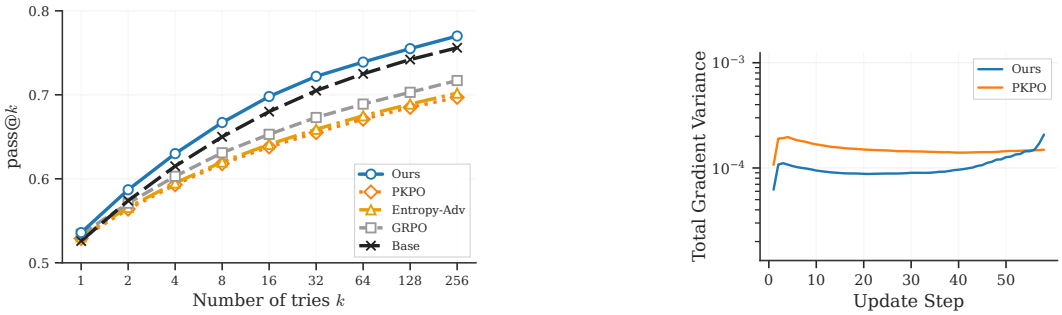
H.3 Code Generation with Continuous Rewards

To evaluate MaxPO beyond mathematical reasoning, we conduct an additional code-generation experiment following the AceCoder setup (Zeng et al., 2025). We train Qwen2.5-7B-Instruct (Yang et al., 2025b) on the 21,262-problem hard split of AceCode-87K. Unlike the binary pass/fail reward of AceCoder, we use the fraction of unit tests passed as a continuous reward $r \in [0, 1]$; note that the unbiasedness and exact-centering properties of L2O are not restricted to binary rewards and apply to general scalar rewards. All methods use $G = 8$, $K = 2$, 60 RL steps, and identical hyperparameters, and we evaluate the final checkpoint of each run. We evaluate unbiased pass@k with 256 samples per problem on seven benchmarks: HumanEval (Chen et al., 2021), HumanEval+ and MBPP+ (Liu et al., 2023), MBPP (Austin et al., 2021), BigCodeBench-Complete-Hard and BigCodeBench-Instruct-Hard (Zhuo et al., 2025), and LiveCodeBench v4 (Jain et al., 2025).

As shown in Table 6, MaxPO outperforms every baseline at every k and also exceeds PKPO on each of the seven individual benchmarks at pass@256. MaxPO also achieves a mean Adam-moment variance proxy (Sec. 5.2.3) of 1.071×10^{-4}, which is 29.3% lower than that of PKPO (1.514×10^{-4}). These results show that the performance and variance-reduction benefits observed in binary-reward mathematical reasoning extend to code generation with continuous rewards.

I Broader Impact

Max@K and pass@K are increasingly important metrics for reasoning models because they capture the quality of a model under repeated sampling, which is a common inference-time protocol. By identifying a canonical finite-batch advantage form for max@K, this work provides a principled foundation for designing, comparing, and diagnosing policy-gradient estimators for such inference-time objectives. This may help improve the efficiency and stability of post-training methods and reduce unnecessary variance in optimization. At the same time, better optimization of repeated-



(a) Task-average pass@k ($k \leq 256$)

(b) Adam-moment variance proxy

Figure 11: Code generation with continuous rewards (Qwen2.5-7B-Instruct). (a) Task-average pass@k over the seven code benchmarks of Table 6. (b) Adam-moment variance proxy (Sec. 5.2.3) during training.

sampling reasoning objectives may strengthen models that can be misused for large-scale cheating, deceptive assistance, or other forms of harmful automated problem solving.