Breaking *Winner-Takes-All*: Cooperative Policy Optimization Improves Diverse LLM Reasoning

Haoxuan Chen[1]* **Tianming Liang**[1]* **Wei-Shi Zheng**[1, 3] **Jian-Fang Hu**[1, 2, 3]†
[1]School of Computer Science and Engineering, Sun Yat-sen University, China
[2]Guangdong Province Key Laboratory of Information Security Technology, China
[3]Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, China
{chenhx253, liangtm}@mail2.sysu.edu.cn, hujf5@mail.sysu.edu.cn

Abstract

Reinforcement learning with verifiers (RLVR) has become a central paradigm for improving LLM reasoning, yet popular group-based optimization algorithms like GRPO often suffer from exploration collapse, where the models prematurely converge on a narrow set of high-scoring patterns, lacking the ability to explore new solutions. Recent efforts attempt to alleviate this by adding entropy regularization or diversity bonus. However, these approaches do not change the *winner-takes-all* nature, where rollouts still compete for individual advantage rather than cooperating for maximizing global diversity. In this work, we propose Group Cooperative Policy Optimization (GCPO), which shifts the training paradigm from rollout competition to team cooperation. Specifically, GCPO replaces independent rollout scoring with team-level credit assignment: a rollout is rewarded by how much it contributes to the team's valid solution coverage, rather than its individual accuracy. This coverage is described as a determinant volume over reward-weighted semantic embeddings, where only correct and non-redundant rollouts contribute to this volume. During advantage estimation, GCPO redistributes the collective team reward to each single rollout according to its average marginal contribution to the team. This cooperative training paradigm routes optimization toward non-redundant correct reasoning paths. Experiments across multiple reasoning benchmarks demonstrate that GCPO significantly improves both reasoning accuracy and solution diversity over existing approaches. Code will be released.

1 Introduction

Reinforcement learning with verified reward (RLVR) has emerged as a powerful paradigm for enhancing LLM reasoning [30, 22]. Group-based policy optimization methods such as GRPO [10] leverage relative comparisons among sampled solutions to improve training stability, making RLVR practical at scale. With outcome-based supervision, RLVR has driven substantial gains across math, coding, logical, and multi-step reasoning tasks, making them as a key component of reasoning-oriented LLM training [7, 33, 41, 42].

Despite these successes, group-based optimization approaches often struggle with exploration collapse: while models improve pass@1 accuracy, they often do so by concentrating probability mass on a narrow set of homogeneous reasoning patterns, lacking the ability to explore new, potentially better reasoning paths [24, 31, 38, 2]. This highlights a central challenge for RLVR: improving reasoning requires not only reinforcing high-score responses but also sustaining exploration across distinct reasoning paths [16, 35].

*Equal contribution.
†Corresponding author.

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

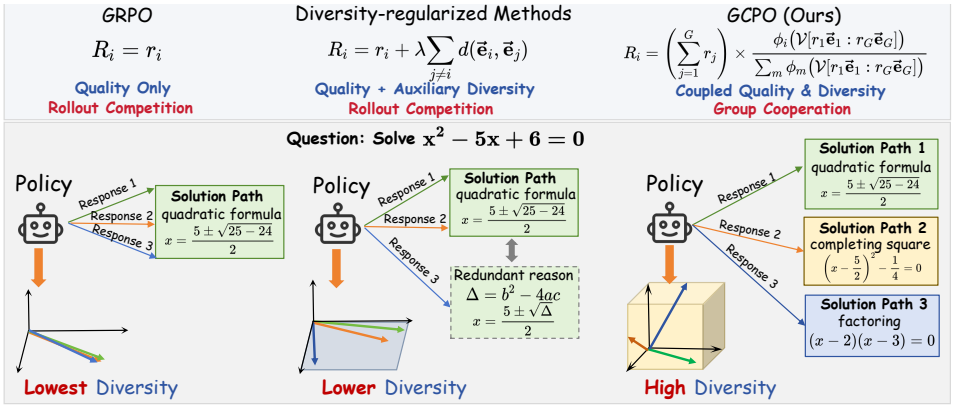

Figure 1: **Comparison of group-based optimization strategies.** *Left*: GRPO optimizes solely for individual correctness, often leading to exploration collapse. *Middle*: Diversity-regularized methods add a diversity bonus to the reward, but this only produces superficial variations of already-successful reasoning paths. *Right*: Our GCPO considers each rollout's marginal contribution to the group's coverage of solutions, incentivizing diverse correct reasoning paths.

Recent works attempt to address this by either adding entropy regularization into the policy distribution [31, 4, 1] or enforcing variation among solutions under global constraints [45, 3, 19, 13]. However, these approaches typically treat diversity term as an additional constraint to the correctness rewards, rather than naturally combining them for cooperative optimization. More importantly, such regularization paradigms do not essentially change the *winner-takes-all* nature: rollouts are still evaluated and optimized by their competitive advantages over others. Under this formulation, models tend to conservatively maximize rewards by generating superficial variations of already-successful solutions, rather than exploring truly diverse reasoning paths, as shown in Figure 1. Consequently, the contradiction between quality and diversity remains largely unresolved in LLM reasoning.

In this work, we propose GCPO (***G**roup **C**ooperative **P**olicy **O**ptimization*), a novel cooperative algorithm for RL-based LLM reasoning. Instead of scoring each rollout independently, GCPO assigns rewards for them through team-level credit assignment. Specifically, GCPO treats a group of sampled rollouts as a cooperative team, and defines its **team value** as the team's coverage of valid reasoning paths. Under this objective, each rollout is rewarded by how much it expands the team's coverage beyond what the others already provide. GCPO instantiates this team value as a determinantal volume, which couples correctness and reasoning diversity. It increases when correct rollouts cover distinct reasoning paths, while incorrect or redundant ones contribute little additional value. To fairly allocate this team value, GCPO redistributes the team's original reward to each rollout according to its average marginal contribution to the team. This team-level cooperative mechanism preserves the task-level reward budget while routing optimization toward non-redundant correct reasoning paths. Extensive experiments across reasoning benchmarks covering multiple domains show that GCPO improves both reasoning accuracy and meaningful solution diversity.

2 Related Work

RLVR. RLVR has become a prominent approach for improving LLM reasoning [6, 10, 26, 15, 20]. While effective, it often suffers from limited exploration, where policies prematurely converge to a narrow set of reasoning patterns [31, 39]. To mitigate this, prior work has introduced unstructured randomness via entropy regularization [12], modified objectives using pass@k-based rewards [4], or intervened in rollout dynamics [2]. However, these methods mainly stimulate exploration by perturbing sampling behavior with exploration-oriented signals. In contrast, GCPO keeps the rollout process simple and introduces cooperation at team-level: sampled rollouts are evaluated as a team, so exploration is guided by their collective coverage rather than isolated stochasticity.

Diversity in LLM Reasoning. Prior studies promote diversity by manipulating policy uncertainty at different levels. Token-level methods primarily rely on entropy bonuses or KL constraints [8, 23, 44, 1, 31, 43] to increase short-term randomness, but fail to explicitly encourage diverse reasoning rollouts. More recent global approaches leverage pass@k signals [4, 35, 21] or promote semantic diversity among candidate solutions [3, 19, 13]. However, these methods usually introduce diversity

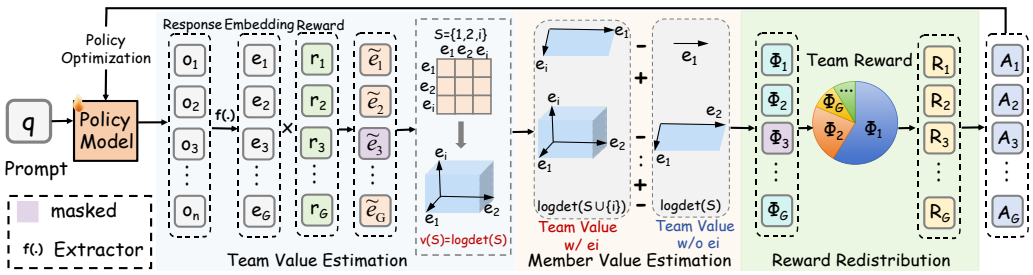

Figure 2: Pipeline of GCPO framework. We formulate the contribution via coupled diversity and quality of response o_i within a group of G rollouts as reward shaper to incentivize deep exploration.

as an auxiliary objective, while leaving rollout-level credit assignment unchanged. In contrast, GCPO incorporates diversity into a group-level value function and assigns credit by each rollout's marginal contribution to reward-gated semantic coverage.

3 GCPO: From Rollout Competition to Team Cooperation

3.1 Preliminaries

Group Relative Policy Optimization. We briefly review GRPO [10], which serves as the optimization backbone for RLVR in reasoning tasks. For each query q, the old policy model $\pi_{\theta_{\text{old}}}$ samples G candidate responses $\{o_1, \ldots, o_G\}$. Each response is assigned a verifiable binary reward $r_i \in \{0, 1\}$ based on matching against the ground-truth answer, which helps mitigate reward hacking [9]. The policy π_θ is updated by maximizing:

$$J_{\text{GRPO}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|q)} \left[\frac{1}{G}\sum_{i=1}^{G}\sum_{t=1}^{|o_i|} \text{CLIP}(\rho_{i,t}, A_i) - \beta\mathbb{KL}(\pi_\theta||\pi_{\text{ref}}) \right], \tag{1}$$

where $\rho_{i,t} = \frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|q,o_{i,<t})}$ is the importance weight. The objective employs PPO-style clipping $\text{CLIP}(\rho_{i,t}, A_i) = \min(\rho_{i,t}A_i, \text{clip}(\rho_{i,t}, 1-\epsilon, 1+\epsilon)A_i)$ for trust-region updates [29], and a KL penalty to constrain deviation from a reference model π_{ref}.

Determinantal Point Processes. Determinantal Point Processes (DPPs) [17] are probabilistic models over subsets that favor diversity, with L-ensembles quantifying subset diversity through determinant-based geometric volume. Let $\mathcal{Y} = \{1, 2, \ldots, N\}$ be a ground set of items, and $\mathbf{L} \in \mathbb{R}^{N\times N}$ be a symmetric, positive semi-definite matrix. An L-ensemble defines a probability measure over random subsets $Y \subseteq \mathcal{Y}$ such that for every subset $A \subseteq \mathcal{Y}$:

$$\Pr(Y = A) \propto \det(\mathbf{L}_A) \tag{2}$$

where $\mathbf{L}_A$ denotes the principal submatrix of $\mathbf{L}$ indexed by A, $\det(\mathbf{L}_A)$ is the determinant of $\mathbf{L}_A$.

Geometrically, if the entries of $\mathbf{L}$ represent inner products of feature vectors, $\det(\mathbf{L}_A)$ corresponds to the squared volume spanned by the items in A. This volume naturally increases when the vectors are diverse and diminishes toward zero when they are similar.

3.2 Group Cooperative Policy Optimization

In this work, we propose GCPO, an algorithm that shifts the *competition* among sampled rollouts to *cooperation*. GCPO treats the rollout group as a cooperative team and rewards each rollout by how much it expands the team's coverage of valid reasoning directions. As illustrated in Figure 2, this is realized through three components: quality-diversity coupled team value estimation (§3.2.1), Shapley-based cooperative member value estimation(§3.2.2), and reward redistribution (§3.2.3).

3.2.1 Team Value Estimation

Here, we introduce a team value function to quantify the collective coverage of valid reasoning paths within a rollout group. The proposed formulation couples two desiderata: **quality**, which measures

the objective correctness of individual rollouts, and **diversity**, which captures the extent to which the team spans distinct and non-redundant semantic directions.

For each response o_i, we extract a semantic representation and weight it by its reward, i.e., $z_i = f_\phi(o_i)$, $\bar{z}_i = \frac{z_i}{\|z_i\|_2}$, $\tilde{z}_i = r_i \bar{z}_i$, where f_ϕ denotes the frozen encoder and r_i is the reward assigned to o_i. We then define a quality-diversity coupled Gram kernel:$L_{ij} = \langle \tilde{z}_i, \tilde{z}_j \rangle = r_i r_j \langle \bar{z}_i, \bar{z}_j \rangle$. This kernel induces a team-level semantic geometry over the responses. It measures the quality-weighted overlap between two reasoning rollouts: high-reward responses contribute strongly, whereas similar responses exhibit redundant coverage. Consequently, correct rollouts with distinct semantic directions enlarge the collective coverage of the team, while redundant ones contribute little.

Inspired by the DPP in §3.1, we define the team value of a rollout set S as the regularized log-determinant volume

$$v(S) = \log\det\left(I_{|S|} + \eta L_S\right). \tag{3}$$

L_S is the submatrix of L indexed by S and $\eta > 0$ controls its contribution. Identity matrix $I_{|S|}$ provides a positive volume and ensures numerical stability when rollouts are highly correlated.

We adopt a determinant-based formulation for the team value because it measures the collective volume spanned by a response set rather than pairwise separations. This is essential for mitigating the limitations of additive diversity bonuses. Pairwise distances do not reflect team-level diversity, since they may remain large even when responses are organized into internally redundant clusters. In contrast, the determinant captures the spectral structure of the Gram matrix, increasing when responses contribute independent directions and saturating when additional rollouts are redundant with the existing set. Proposition 1 formalizes these properties.

Proposition 1. *Let $\lambda_1, \ldots, \lambda_{|S|}$ be the eigenvalues of L_S. Then the team value can be expressed as:*

$$v(S) = \sum_{\ell=1}^{|S|} \log(1 + \eta\lambda_\ell). \tag{4}$$

This form shows how the value function balances correctness and diversity. Zero-reward rollouts do not contribute to the team value. Moreover, m mutually orthogonal correct responses yield $v(S) = m\log(1+\eta)$, whereas m identical correct responses yield $v(S) = \log(1+\eta m)$. Therefore, the value grows more rapidly when correct responses span independent semantic directions, while it saturates when correct responses are redundant. Thus, **diverse correct reasoning paths are assigned higher team value, whereas redundant correct paths yield diminishing returns.**

3.2.2 Member Value Estimation

The team value measures the collective coverage of a coalition, while policy optimization requires a per-rollout credit that reflects each member's contribution to the group. This credit is derived in two steps: each rollout's **marginal contribution** to a coalition is first computed, and these context-dependent contributions are then aggregated into a fair **cooperative credit** via Shapley values.

Marginal contribution. Given a rollout i and coalition S that does not contain i, the marginal contribution of i is defined as the additional team value induced by adding i to S:

$$\Delta_i(S) = v(S \cup \{i\}) - v(S). \tag{5}$$

Theorem 2. *Let $\tilde{Z}_S \in \mathbb{R}^{|S|\times d}$ denote the representation matrix of coalition S, whose rows are given by $\{\tilde{z}_j\}_{j\in S}$. For any $i \notin S$, the marginal contribution can be calculated as:*

$$\Delta_i(S) = \log\left(1 + \eta r_i^2 \bar{z}_i^\top \left(I_d + \eta \tilde{Z}_S^\top \tilde{Z}_S\right)^{-1} \bar{z}_i\right), \tag{6}$$

where r_i is the reward of i. The marginal contribution is bounded as $0 \le \Delta_i(S) \le \log(1+\eta r_i^2)$.

Here, r_i^2 gates the contribution and the quadratic form measures the novelty of z_i after accounting for directions already covered by S. Theorem 2 confirms that **a rollout is valuable only when it is both correct and semantically non-redundant.** Moreover, it is upper-bounded, ensuring that the credit assigned to any rollout remains **well-controlled during policy optimization.**

Cooperative credits. The marginal contribution $\Delta_i(S)$ is coalition-dependent, and thus does not provide a unique rollout-level credit by itself. A straightforward choice is the leave-one-out (LOO)

score $\Delta_i([G]\setminus\{i\})$, which fixes S to the rest of the full group. But LOO suffers from mutual masking: when multiple rollouts instantiate the same valid reasoning direction, removing any single one has only a limited effect on the total volume. As a result, all such rollouts may receive small individual credits, despite collectively contributing a valuable reasoning direction.

To obtain a fair and context-independent credit for each rollout, we use the Shapley value [34], which attributes the team value by averaging each rollout's marginal contribution across all possible coalition formation orders:

$$\phi_i = \sum_{S\subseteq[G]\setminus\{i\}} \frac{|S|!(G-|S|-1)!}{G!}\left[v(S\cup\{i\})-v(S)\right]. \tag{7}$$

The Shapley value provides a principled way to convert coalition-dependent marginal contributions into a coalition-independent cooperative credit and mitigate the risk of mutual masking.

Theorem 3. *Consider a set $C=\{1,\dots,M\}$ of m verifier-correct rollouts with identical semantic embeddings, i.e., $\tilde{z}_i = u$ for all $i\in C$. Then the Shapley value of each rollout is*

$$\phi_i = \frac{1}{M}\log\left(1+\eta M\|u\|_2^2\right), \qquad i\in C. \tag{8}$$

Theorem 3 highlights a key distinction between GCPO and LOO-based credit assignment. LOO measures the decrease in team value after removing a rollout from the full group. As a result, when multiple rollouts realize the same valid reasoning path, the remaining rollouts can compensate for the removed one, causing LOO to underestimate their shared contribution. **Shapley instead divides the team value equally among all rollouts that realize the same reasoning path.**

Since rollouts with zero reward have zero marginal contribution according to Theorem 2, they do not affect the Shapley allocation. Let $\mathcal{P}=\{i\in[G]\mid r_i>0\}$ be the set of positive-reward rollouts. In practice, we therefore efficiently compute Shapley values only over $\mathcal{P}$:

$$\phi_i = \sum_{S\subseteq\mathcal{P}\setminus\{i\}} \frac{|S|!(|\mathcal{P}|-|S|-1)!}{|\mathcal{P}|!}\left[v(S\cup\{i\})-v(S)\right]. \tag{9}$$

3.2.3 Reward Redistribution

After computing cooperative values for individual rollouts, GCPO converts them into training signals by reward redistribution. Rather than adding these values as extra bonuses, which would increase the total reward mass and may encourage reward hacking, GCPO keeps the total batch reward fixed and reallocates it to rollouts in proportion to their cooperative credits. Specifically, let $R_{\text{batch}}=\sum_{i=1}^{G} r_i$, the final reward of GCPO is defined as:

$$\tilde{r}_i = \begin{cases} R_{\text{batch}}\cdot\dfrac{\phi_i}{\sum_{j\in\mathcal{P}}\phi_j}, & r_i>0,\\ 0, & r_i=0. \end{cases} \tag{10}$$

Since $\sum_{i=1}^{G}\tilde{r}_i = R_{\text{batch}} = \sum_{i=1}^{G} r_i$, the total reward in the batch remains unchanged. This conservation property ensures that GCPO does not introduce additional reward mass: it preserves the task-level reward budget while reallocating credit among correct rollouts, thereby reducing the risk of reward hacking associated with additive diversity bonuses.

Proposition 4. *For a query q, let $B=\{o_i\}_{i=1}^{G}\sim\pi(\cdot\mid q)^G$ be a group of i.i.d. responses with verifier rewards $r_i=r(o_i)$. Then the expected batch-average reward of GCPO is preserved: $J^{\Sigma}_{\text{GCPO}}(\pi;q) = \mathbb{E}_{B\sim\pi^G}\left[\frac{1}{G}\sum_{i=1}^{G}\tilde{r}_i\right] = \mathbb{E}_{o\sim\pi(\cdot|q)}[r(o)] = J_{\text{obj}}(\pi;q)$. In contrast, for an additive diversity-shaped reward $r_i^{\text{add}} = r_i + \lambda d_i$, we have: $J_{\text{add}}(\pi;q) = J_{\text{obj}}(\pi;q) + \lambda\,\mathbb{E}_{B\sim\pi^G}\left[\frac{1}{G}\sum_{i=1}^{G} d_i\right]$.*

Proposition 4 distinguishes GCPO from additive diversity shaping at the reward-budget level. Additive shaping increases the expected reward by an external diversity term, which may favor diversity even when it is not aligned with correctness. In contrast, GCPO conserves the group-level reward and only reallocates it within the response group. Thus, diversity affects learning only through credit assigned to correct, non-redundant rollouts, rather than through additional reward mass.

3.2.4 GCPO Objective

GCPO can be seamlessly incorporated into GRPO by modifying only the reward used for advantage estimation. Specifically, we replace the original reward r_i with the redistributed reward $\tilde{r}_i$, leading to $A_i^{GCPO} = (\tilde{r}_i - \mu_{\tilde{r}})/(\sigma_{\tilde{r}} + \epsilon_{num})$. The policy is optimized using standard clipped surrogate loss:

$$\mathcal{L}_{GCPO}(\theta) = \mathbb{E}_{x\sim\mathcal{D}}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{T_i}\sum_{t=1}^{T_i}\min\left(\rho_{i,t}A_i^{GCPO}, \text{clip}(\rho_{i,t}, 1-\epsilon_c, 1+\epsilon_c)A_i^{GCPO}\right)\right] - \beta\mathbb{D}_{KL}$$

The encoder f_ϕ, Shapley credits ϕ_i, and redistributed rewards $\tilde{r}_i$ are treated as stop-gradient quantities during policy optimization. Thus, GCPO modifies the training signal only through the advantages induced by reward redistribution. We make this effect explicit through the following decomposition.

Proposition 5. *Let $\Delta A_i \triangleq A_i^{GCPO} - A_i^{GRPO}$ denote the advantage correction introduced by GCPO for the i-th rollout. The unclipped policy-gradient surrogate admits the decomposition*

$$\nabla_\theta\mathcal{L}_{GCPO} = \nabla_\theta\mathcal{L}_{GRPO} + \mathbb{E}_{x\sim\mathcal{D}}\left[\frac{1}{G}\sum_{i=1}^{G}\nabla_\theta \log \pi_\theta(o_i \mid x)\cdot\Delta A_i\right]. \tag{11}$$

Proposition 5 reveals GCPO's optimization signal. The first term recovers the standard GRPO reward update, while the second term acts as an internal credit-routing correction induced by reward redistribution. This correction favors semantically complementary correct rollouts over redundant ones, thereby routing exploration within the valid solution space without promoting incorrect rollouts.

Implementation. In practice, GCPO incurs only a modest computational overhead. Although the coupled kernel is defined over all rollouts, rollouts with zero reward have zero contribution and can be excluded from Shapley computation. Thus, Shapley values can be computed only over $\mathcal{P}$ using Eq. (9). For the rollout group sizes $G \leq 8$, exact enumeration over correct subsets is feasible. For larger groups, we estimate Shapley value by Monte Carlo permutations detailed in Appendix A.3.

4 Experiments

4.1 Experimental Setup

Implementation Details. We conduct all experiments on the filtered DAPO-Math-17K [14], which removes duplicated samples, and train base models with increasing capacity: Qwen3-1.7B and 4B [41]. We adopt all-MiniLM-L6-v2 [27] as a lightweight sentence encoder for extracting semantic response embeddings. All methods use the same training configuration: batch size 256, $G = 8$ rollouts during policy optimization, learning rate $1e^{-6}$, and maximum response length 8192.

Evaluation Setting. Evaluation is conducted on a diverse set of benchmarks spanning math reasoning and broader reasoning domains. For in-domain evaluation, we use AIME2024 [46], AIME2025, MATH500 [6], Minerva Math [18], and Olympiad [11]. For out-of-domain generalization, we evaluate on ARC-Challenge [5], GPQA [28], and MMLU-Pro [37], covering multi-domain reasoning. For smaller test sets(AIME 24/25), we report **Avg@16**, computed by averaging correctness over 16 independent rollouts at temperature 0.7, and **Pass@1**, computed from a single rollout per problem for others. In addition, we evaluate **Pass@k** for $k \in \{2, 4, \ldots, 64\}$ by sampling 64 independent rollouts per problem. We compare GCPO against existing sota optimization algorithms like GRPO [30], DAPO [44], DIVER [13], DQO [3], as well as the corresponding base models.

4.2 Main Results

Comparison with RLVR Baselines. Table 1 reports results on five in-domain reasoning benchmarks and three out-of-domain reasoning benchmarks across two model scales. GCPO consistently improves Pass@1 over standard RLVR baselines. On Qwen3-1.7B, GCPO achieves an in-domain average of 35.3, with large gains on challenging benchmarks such as AIME24 and Olympiad. The trend remains at the larger scale: on Qwen3-4B, GCPO reaches 46.9, outperforming DAPO by +8.0 points. These results suggest that GCPO improves accuracy while maintaining broader coverage of reasoning paths.

Comparison with Diversity-Aware Methods. GCPO also outperforms diversity-aware baselines across most evaluated settings. On Qwen3-1.7B, GCPO improves the in-domain average from DQO's 32.9 to 35.3, and raises the OOD average to 52.0. On Qwen3-4B, GCPO further improves the

Table 1: Comparison on in and out-of-domain tasks. **Bold**/underline denote best/second-best.

Methods	In-Domain Performance						Out-of-Domain Performance			
	AIME24	AIME25	MATH500	Minerva	Olympiad	Avg.	ARC-c	GPQA	MMLU-Pro	Avg.
Qwen3-1.7B(Base)	13.3	10.0	57.6	17.3	23.7	24.4	54.1	20.2	27.5	33.9
GRPO [10]	16.7	20.0	57.4	18.4	29.5	28.4	75.6	24.3	38.5	46.1
DAPO [44]	20.0	10.0	63.8	21.3	32.9	29.6	**79.4**	28.2	41.0	49.5
Div-R1 [43]	22.1	15.3	60.2	19.5	31.2	29.7	77.8	25.1	39.9	47.6
DIVER [13]	25.4	17.6	61.5	20.8	33.5	31.8	78.3	30.2	43.8	50.8
DQO [3]	27.5	18.4	62.3	22.1	34.2	32.9	76.4	28.3	41.5	48.7
GCPO	**30.7**	**21.2**	**64.7**	**23.5**	**36.2**	**35.3**	79.1	**31.8**	**45.2**	**52.0**
Qwen3-4B(Base)	13.3	16.7	64.8	26.1	32.8	30.7	66.9	26.3	30.9	41.4
GRPO [10]	36.7	33.3	64.0	25.0	37.5	39.3	84.6	44.4	60.1	63.0
DAPO [44]	33.3	40.0	59.0	25.7	36.7	38.9	89.6	46.0	62.8	66.1
Div-R1 [43]	38.5	42.5	61.5	26.2	37.9	41.3	86.9	45.2	61.5	64.7
DIVER [13]	42.3	44.2	62.8	26.4	38.5	42.8	88.4	**48.5**	62.0	66.3
DQO [3]	46.8	46.5	63.7	26.0	39.2	44.4	86.5	47.2	59.2	64.3
GCPO	**51.7**	**49.0**	**66.4**	**26.8**	**40.4**	**46.9**	**91.4**	47.5	**63.8**	**67.6**

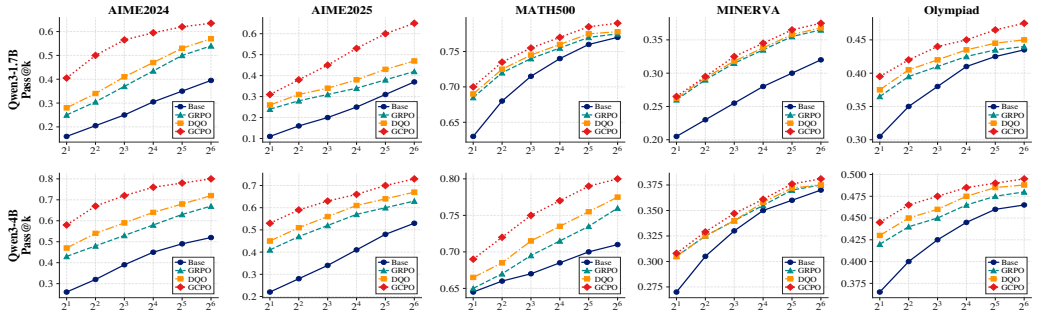

Figure 3: Pass@k performance across five benchmarks for both Qwen3-1.7B and Qwen3-4B. Base models serve as backbones. GCPO consistently outperforms other models across all values of k.

in-domain average over DQO by +2.5 points and achieves the best OOD average of 67.6. These gains suggest that treating diversity as cooperative credit assignment is more effective than additive diversity, especially on harder benchmarks where redundant patterns can limit exploration.

Out-of-Domain Generalization. As shown in Table 1, GCPO achieves the highest average OOD accuracy across both model scales. The performance gains are particularly striking on knowledge-intensive benchmarks: On GPQA, GCPO outperforms DAPO by 3.6 points and DIVER by 1.6 points on Qwen3-1.7B. This result demonstrates that the cooperative mechanism effectively mitigates task-specific pattern collapse and cultivates a robust reasoning capability that generalizes well beyond the primary mathematical optimization objective.

4.3 Diversity Analysis

Reasoning Capacity Upper Bound. We evaluate Pass@k, the probability of obtaining at least one correct solution from k independent samples, as an empirical measure of the model's solution discovery capacity. As shown in Figure 3, GCPO consistently outperforms GRPO, DIVER, and DQO across k at both model scales. On challenging benchmarks such as AIME2024 and AIME2025, the performance gap increases with larger k, suggesting that GCPO expands the coverage of team solution paths rather than only improving individual samples. On MATH500, where baseline accuracy is already high, GCPO still provides stable gains across the full k range, indicating that its benefit is not limited to low-accuracy regimes. Overall, these results show that GCPO improves exploratory capacity while maintaining solution quality.

Multi-dimensional Diversity Characterization. Figure 4(a) evaluates solution diversity on MATH500 using five metrics: self-ROUGE, self-BLEU, Distinct-1, LLM-as-Judge, and Pass@16, with details provided in Appendix A. GCPO shows consistent improvements across all metrics. The lexical-level metrics indicate reduced surface redundancy, while LLM-as-Judge and Pass@16 capture semantic reasoning diversity and valid solution coverage. Their joint improvement suggests that the

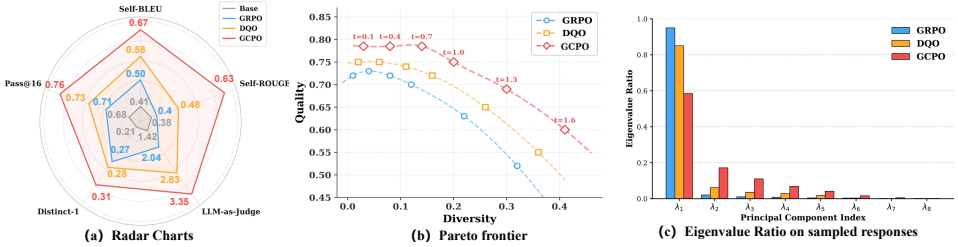

Figure 4: **Diversity analysis.** (a) Radar chart comparing five diversity metrics across methods. (b) Pareto frontier at varying sampling temperatures. (c) Eigenvalue Ratio of sampled responses.

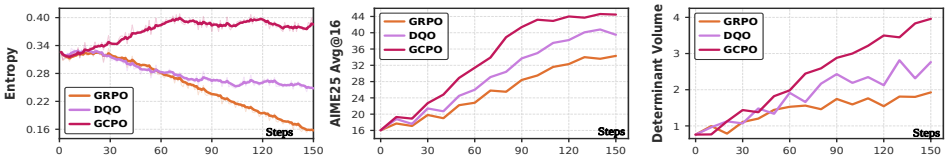

Figure 5: **Training dynamics on Qwen3-4B.** From left to right: policy entropy, AIME2025 Avg@16, and the sample rollouts' determinantal team value in Eq. (3).

determinantal objective encourages meaningful diversity among correct solutions, rather than merely increasing superficial token-level variation.

Quality-Diversity Trade-off. Figure 4(b) studies the trade-off between reasoning quality and solution diversity by varying the sampling temperature t, where quality is measured by Pass@1 and diversity by LLM-as-Judge. As expected, higher temperatures generally increase diversity at the cost of accuracy. Across this trade-off, GCPO consistently lies closer to the upper-right region than GRPO and DQO, achieving higher accuracy at comparable diversity levels. This suggests that GCPO encourages diversity that is better aligned with correctness, rather than relying on high-temperature sampling noise. The result is consistent with our cooperative reward redistribution, which promotes non-redundant valid reasoning paths while preserving the reward budget.

Diversity Collapse Analysis. Here, we explore the mechanism behind diversity collapse by eigenvalue analysis. Figure 4(c) reports the eigenspectrum of the response Gram matrix. When sampled responses concentrate on a single reasoning mode, the leading eigenvalue dominates and the remaining spectrum decays rapidly. GRPO exhibits this pattern, indicating that its responses are concentrated in a narrow semantic subspace. In contrast, GCPO has a lower leading eigenvalue ratio and a flatter spectral profile, suggesting that its responses span more independent semantic directions. This observation is consistent with the team value in Eq. (3), which favors correct responses that expand the covered reasoning subspace. To further understand how such diversity collapse emerges during optimization, Figure 5 tracks policy entropy, Avg@16, and determinantal volume over training. GRPO and DQO quickly lose exploration capacity: their entropy decreases or plateaus, and their determinantal volume saturates early, indicating that additive bonus mostly reinforce already covered reasoning patterns. In contrast, GCPO maintains higher but controlled entropy, while its determinantal volume continues to grow throughout training. This suggests that GCPO sustains exploration over non-redundant correct reasoning paths. The simultaneous increase in Avg@16 further shows that this expanded coverage translates into improved reasoning performance rather than superficial variation.

Qualiative Analysis. Figure 6 visualizes the reasoning strategies discovered by different methods. Solutions generated by GRPO exhibit pattern collapse, narrowly clustering around standard, low-complexity algebraic solution. In contrast, GCPO expands the solution space along both the complexity and insight axes. It successfully navigates towards high-insight reasoning paths, such as trail/flow viewpoints, which remain unexplored by GRPO, demonstrating that GCPO constructs a richer, multi-modal reasoning model. Please refer to Appendix E.3 for more qualitative results.

4.4 Ablation Studies

Table 2 ablates three design hypotheses behind GCPO: quality-diversity coupling, volume-based team coverage, and cooperative reward redistribution.

Impact of Quality-Diversity Coupling. Removing the quality-diversity coupling causes the largest performance drop, reducing Pass@1/Avg@16 from 49.02/45.46 to 43.60/39.95. Notably, the LLM-

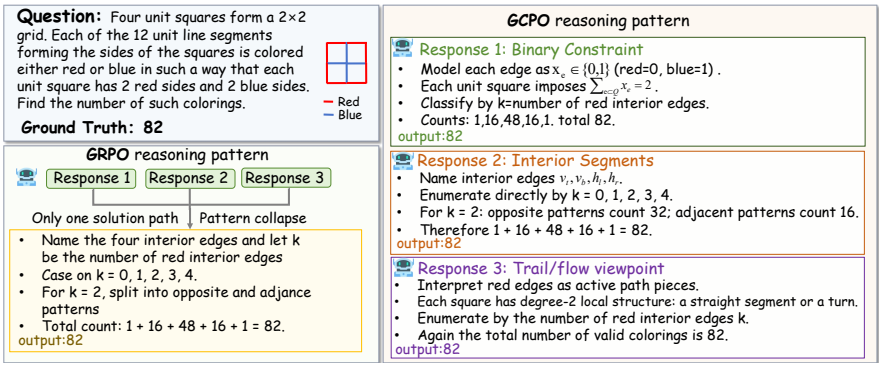

Figure 6: Qualitative Diversity Analysis of GRPO and GCPO reasoning patterns.

Table 2: Ablation study on the core components of GCPO. **Bold** indicates the best performance.

Model Variant	Quality		Diversity	
	Pass@1 (%)	Avg@16	Pass@16 (%)	LLM-as-Judge
GCPO	**49.02**	**45.46**	**66.47**	**2.58**
w/o Quality-Diversity Coupling	43.60	39.95	60.73	2.47
w/o DPP Volume	46.95	43.25	62.84	1.92
w/o Shapley Value	48.23	44.50	64.05	2.17
Additive Bonus	48.17	44.27	64.79	2.33

as-Judge diversity score remains relatively high, indicating that diversity alone is insufficient when it is not tied to correctness. This supports our reward-gated formulation, where only positive-reward rollouts can contribute to the team value.

Impact of Determinantal Team Volume. Removing the DPP-based team volume mainly weakens diversity-related metrics, decreasing LLM-as-Judge from 2.58 to 1.92 and Pass@16 from 66.47 to 62.84. This suggests that local or pairwise diversity signals are less effective at modeling group-level coverage. The log-determinant volume provides a stronger structural signal by assigning diminishing value to redundant rollouts and favoring responses that expand independent reasoning directions.

Impact of Cooperative Credit and Reward Redistribution. Replacing Shapley credit with a leave-one-out estimator degrades both accuracy and diversity, showing the importance of fair credit allocation among rollouts that share similar reasoning paths. This is consistent with our analysis that LOO can underestimate the shared value of duplicated discoveries, whereas Shapley distributes the team value across all contributors. In addition, the additive-bonus variant underperforms GCPO, indicating that simply adding diversity reward is less effective than redistributing the original verifier reward. Together, these results show that GCPO's gains come from the combination of correctness-gated coverage, determinantal team value, and cooperative reward redistribution.

Computational Overhead Analysis. Table 3 reports the per-step runtime of different optimization methods. GCPO requires 320.6s per step, introducing only a 3.7s overhead over GRPO and a runtime comparable to DIVER [13]. Meanwhile, GCPO achieves a Pass@1 accuracy of 49.0% on AIME25, improving over GRPO by 15.7 points. These results indicate that our cooperative mechanism introduces limited computational overhead while providing substantial gains in reasoning performance.

Table 3: Computation overhead analysis.

Method	Time/step(s)	Acc.(%)
GRPO	316.9	33.3
DIVER	320.5 (+3.6)	44.2 (+10.9)
GCPO	320.6 (+3.7)	49.0 (+15.7)

5 Conclusion

We introduced GCPO, a cooperative policy optimization framework for improving diverse exploration and mitigating reasoning-pattern collapse in RLVR. GCPO evaluates each rollout by its contribution to the group's coverage of valid reasoning paths, rather than by standalone quality or diversity. It operationalizes this through a reward-weighted determinantal team value, Shapley-based cooperative member value estimation, and redistribution of the original reward. This design preserves the task-level reward budget while assigning preference signals to non-redundant correct rollouts. Experiments across benchmarks and model scales show that GCPO improves both reasoning accuracy and solution

diversity. A key limitation lies in the cost of determinant evaluation and Shapley enumeration for larger groups, which motivates future work on scalable approximations and extensions to broader reasoning and open-ended generation tasks.

Acknowledgement

This work was partially supported by NSFC (U22A2095, 62476296), Guangdong Natural Science Funds Project (2023B1515040025, 2022B1515020009), Guangdong Provincial Science and Technology Program Project (2024A1111120017), and Guangdong Provincial Key Laboratory of Information Security Technology (2023B1212060026).

References

[1] Shivam Agarwal, Zimin Zhang, Lifan Yuan, Jiawei Han, and Hao Peng. The unreasonable effectiveness of entropy minimization in llm reasoning. *arXiv preprint arXiv:2505.15134*, 2025.

[2] Liang Chen, Xueting Han, Qizhou Wang, Bo Han, Jing Bai, Hinrich Schutze, and Kam-Fai Wong. Eepo: Exploration-enhanced policy optimization via sample-then-forget. *arXiv preprint arXiv:2510.05837*, 2025.

[3] Yilei Chen, Souradip Chakraborty, Lorenz Wolf, Yannis Paschalidis, and Aldo Pacchiano. Post-training large language models for diverse high-quality responses. *arXiv preprint arXiv:2509.04784*, 2025.

[4] 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*, 2025.

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

[6] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[7] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

[8] 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.

[9] Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. PMLR, 2023.

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

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

[12] Zhenyu Hou, Xin Lv, Rui Lu, Jiajie Zhang, Yujiang Li, Zijun Yao, Juanzi Li, Jie Tang, and Yuxiao Dong. Advancing language model reasoning through reinforcement learning and inference scaling. *arXiv preprint arXiv:2501.11651*, 2025.

[13] Zican Hu, Shilin Zhang, Yafu Li, Jianhao Yan, Xuyang Hu, Leyang Cui, Xiaoye Qu, Chunlin Chen, Yu Cheng, and Zhi Wang. Diversity-incentivized exploration for versatile reasoning. *arXiv preprint arXiv:2509.26209*, 2025.

[14] Hugging Face. Open r1: A fully open reproduction of deepseek-r1, January 2025.

[15] Haichao Jiang, Tianming Liang, Wei-Shi Zheng, and Jian-Fang Hu. Refer-agent: A collaborative multi-agent system with reasoning and reflection for referring video object segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2026, Denver, CO, USA, June 3-7, 2026*, pages 39508–39517. Computer Vision Foundation, 2026.

[16] Yuhua Jiang, Jiawei Huang, Yufeng Yuan, Xin Mao, Yu Yue, Qianchuan Zhao, and Lin Yan. Risk-sensitive rl for alleviating exploration dilemmas in large language models. *arXiv preprint arXiv:2509.24261*, 2025.

[17] Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2-3):123–286, 2012.

[18] Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857, 2022.

[19] 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.

[20] Tianming Liang, Qirui Du, Jian-Fang Hu, Haichao Jiang, Zicheng Lin, and Wei-Shi Zheng. Seg-research: Segmentation with interleaved reasoning and external search. *arXiv preprint arXiv:2602.04454*, 2026.

[21] Tianming Liang, Kun-Yu Lin, Chaolei Tan, Jianguo Zhang, Wei-Shi Zheng, and Jian-Fang Hu. Referdino: Referring video object segmentation with visual grounding foundations. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 20009–20019. IEEE, 2025.

[22] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

[23] Jia Liu, ChangYi He, YingQiao Lin, MingMin Yang, FeiYang Shen, and ShaoGuo Liu. Ettrl: Balancing exploration and exploitation in llm test-time reinforcement learning via entropy mechanism. *arXiv preprint arXiv:2508.11356*, 2025.

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

[25] Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. In *Findings of the association for computational linguistics: ACL 2022*, pages 1864–1874, 2022.

[26] OpenAI. Learning to reason with llms. `https://openai.com/index/learning-to-reason-with-llms`, 2024.

[27] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 3982–3992, 2019.

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

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

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

[31] Han Shen. On entropy control in llm-rl algorithms. *arXiv preprint arXiv:2509.03493*, 2025.

[32] 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. In *Proceedings of the Twentieth European Conference on Computer Systems*, pages 1279–1297, 2025.

[33] Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

[34] Mukund Sundararajan and Amir Najmi. The many shapley values for model explanation. In *International conference on machine learning*, pages 9269–9278. PMLR, 2020.

[35] Christian Walder and Deep Karkhanis. Pass@ k policy optimization: Solving harder reinforcement learning problems. *arXiv preprint arXiv:2505.15201*, 2025.

[36] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022.

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

[38] 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.

[39] Ting Wu, Xuefeng Li, and Pengfei Liu. Progress or regress? self-improvement reversal in post-training. *arXiv preprint arXiv:2407.05013*, 2024.

[40] Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pages 641–649, 2024.

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

[42] Yuting Yang, Haichao Jiang, Tianming Liang, Quan Zhang, and Jian-Fang Hu. Don't guess, just ask: Resolving ambiguity in referring segmentation via multi-turn clarification. *arXiv preprint arXiv:2605.17531*, 2026.

[43] Jian Yao, Ran Cheng, Xingyu Wu, Jibin Wu, and Kay Chen Tan. Diversity-aware policy optimization for large language model reasoning. *arXiv preprint arXiv:2505.23433*, 2025.

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

[45] Qingyang Zhang, Haitao Wu, Changqing Zhang, Peilin Zhao, and Yatao Bian. Right question is already half the answer: Fully unsupervised llm reasoning incentivization. *arXiv preprint arXiv:2504.05812*, 2025.

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

A Implementation Details

A.1 Detailed Settings

We provide additional experimental details in Section 4. All models are trained using the VERL framework [32] and deployed on 8× NVIDIA 5880 Ada Generation GPUs. Table 4 and Table 5 summarize the training and evaluation hyperparameters. Unless otherwise specified, we adopt a rollout group size of $n = 8$, a learning rate of 1×10^{-6}, and binary verifier rewards.

Table 4: Summary of training details.

Training Settings	
Hardware	8×5880 Ada Generation GPUs
Base models	Qwen3-1.7B, 4B
Training dataset	open-r1/DAPO-Math-17K-Processed
Max prompt length	1024
Max response length	4096 / 8192
Gpu memory utilization	0.8
Train batch / mini-batch size	256 / 16
Use dynamic batch size	True
Validate batch size	512
Rollout group size n	8
Learning rate	1×10^{-6}
Temperature (training)	1.0
Clip range $(\epsilon_{\text{low}}, \epsilon_{\text{high}})$	(0.2, 0.28) / (0.2, 0.2) for GRPO
Reward type	Binary reward
Team value factor η	1.0

Table 5: Evaluation settings.

Evaluation settings	
Max response length	4096 / 8192
Top-p (eval)	0.9
Temperature (eval)	0.1 for Pass@1; 0.7 for Avg@16

A.2 Detailed Algorithm

Here, we provide detailed algorithm for Section 3.2. As shown in Alg 1, this cooperative framework ensures that rollouts contributing unique semantic directions to the team receive higher credit, while redundant responses are penalized, all while preserving the total reward budget.

A.3 Shapley Estimation for Large Rollout Groups

Exact Shapley computation requires summing over exponentially many coalitions. In our main experiments, this cost is small because the rollout group size is moderate and Shapley values are computed only over verifier-correct rollouts. For larger rollout groups, we use an unbiased permutation estimator with incremental log-determinant updates. Let $P = \{i \in [G] \mid r_i > 0\}$ be the set of positive-reward rollouts. By Theorem 2, any rollout with $r_i = 0$ has zero marginal contribution to every coalition. Hence Shapley computation can be restricted to P:

$$\phi_i = \sum_{S \subseteq P \setminus \{i\}} \frac{|S|!(m - |S| - 1)!}{m!} \left[v(S \cup \{i\}) - v(S) \right], \qquad i \in P.$$

This is still exponential in m. We now describe a scalable estimator for large m.

Permutation estimator. Let $\Pi(P)$ denote the set of all permutations of P. For a permutation $\pi \in \Pi(P)$, let S_i^π be the set of rollouts that appear before i in π. The Shapley value admits the

Algorithm 1 GCPO reward construction for one rollout group

Input: Responses $\{o_i\}_{i=1}^{G}$, verifier rewards $\{r_i\}_{i=1}^{G}$, encoder f_ϕ, scale η, numerical constant ϵ_{num}
Output: Redistributed rewards $\{\tilde{r}_i\}_{i=1}^{G}$

$\mathcal{P} \leftarrow \{i \in [G] \mid r_i > 0\}, \quad R_{\text{batch}} \leftarrow \sum_{i=1}^{G} r_i$
if $\mathcal{P} = \emptyset$ **then**
    Set $\tilde{r}_i = 0$ and $A_i^{\text{GCPO}} = 0$ for all i, then stop
end if
Stage 1: Calculate Team Value
for $i = 1$ to G **do**
    $z_i \leftarrow f_\phi(o_i)$
    $\bar{z}_i \leftarrow z_i / (\|z_i\|_2 + \epsilon_{\text{num}})$
    $\tilde{z}_i \leftarrow r_i \, \bar{z}_i$
end for
$L \leftarrow [\langle \tilde{z}_i, \tilde{z}_j \rangle]_{i,j=1}^{G}$
Define $v(\emptyset) \leftarrow 0$ and $v(S) \leftarrow \log\det(I_{|S|} + \eta L_S)$ for any non-empty $S \subseteq \mathcal{P}$
Stage 2: Assign Member Value
$m \leftarrow |\mathcal{P}|$
for $i = 1$ to G **do**
    $\phi_i \leftarrow 0$
end for
for each $i \in \mathcal{P}$ **do**
    for each subset $S \subseteq \mathcal{P} \setminus \{i\}$ **do**
        $w(S) \leftarrow \dfrac{|S|!(m - |S| - 1)!}{m!}$
        $\phi_i \leftarrow \phi_i + w(S)\big[v(S \cup \{i\}) - v(S)\big]$
    end for
end for
Stage 3: Redistribute reward
$\Phi \leftarrow \sum_{j \in \mathcal{P}} \phi_j$
for $i = 1$ to G **do**
    if $i \in \mathcal{P}$ and $\Phi > 0$ **then**
        $\tilde{r}_i \leftarrow R_{\text{batch}} \cdot \phi_i / \Phi$
    else
        $\tilde{r}_i \leftarrow 0$
    end if
end for

equivalent form

$$\phi_i = \mathbb{E}_{\pi \sim \text{Unif}(\Pi(P))} \left[v(S_i^\pi \cup \{i\}) - v(S_i^\pi) \right].$$

To see this, fix any coalition $S \subseteq P \setminus \{i\}$. The event $S_i^\pi = S$ occurs when the elements of S appear before i, the elements of $P \setminus (S \cup \{i\})$ appear after i, and the internal orders within the two groups are arbitrary. Thus,

$$\Pr(S_i^\pi = S) = \frac{|S|!(m - |S| - 1)!}{m!},$$

which is exactly the Shapley coefficient.

We sample K independent permutations $\{\pi_k\}_{k=1}^{K}$ and estimate $\widehat{\phi}_i$ with

$$\widehat{\phi}_i = \frac{1}{K} \sum_{k=1}^{K} \left[v(S_i^{\pi_k} \cup \{i\}) - v(S_i^{\pi_k}) \right].$$

This estimator is unbiased: $\mathbb{E}[\widehat{\phi}_i] = \phi_i$.

Incremental log-determinant update. A direct implementation would recompute a log-determinant for every marginal contribution. Instead, we compute all marginal contributions along a sampled permutation incrementally.

For a coalition S, define

$$\mathbf{K}_S = \mathbf{I}_{|S|} + \eta \mathbf{L}_S.$$

Suppose we add a rollout $i \notin S$. After ordering the indices as (S, i),

$$\mathbf{K}_{S\cup\{i\}} = \begin{bmatrix} \mathbf{K}_S & \eta \mathbf{L}_{S,i} \\ \eta \mathbf{L}_{i,S} & 1 + \eta L_{ii} \end{bmatrix}.$$

By the Schur complement,

$$\det(\mathbf{K}_{S\cup\{i\}}) = \det(\mathbf{K}_S) \left(1 + \eta L_{ii} - \eta^2 \mathbf{L}_{i,S} \mathbf{K}_S^{-1} \mathbf{L}_{S,i}\right).$$

Therefore,

$$v(S \cup \{i\}) - v(S) = \log \left(1 + \eta L_{ii} - \eta^2 \mathbf{L}_{i,S} \mathbf{K}_S^{-1} \mathbf{L}_{S,i}\right).$$

In practice, we maintain the Cholesky factor

$$\mathbf{K}_S = \mathbf{C}_S \mathbf{C}_S^\top.$$

Let $\mathbf{b}_{S,i} = \eta \mathbf{L}_{S,i}, a_i = 1 + \eta L_{ii}$. Solving $\mathbf{C}_S \mathbf{y} = \mathbf{b}_{S,i}$ gives:

$$v(S \cup \{i\}) - v(S) = \log s_i, \qquad s_i = a_i - \|\mathbf{y}\|_2^2.$$

The Cholesky factor is then updated as

$$\mathbf{C}_{S\cup\{i\}} = \begin{bmatrix} \mathbf{C}_S & \mathbf{0} \\ \mathbf{y}^\top & \sqrt{s_i} \end{bmatrix}.$$

For the empty coalition, this reduces to

$$v(\{i\}) - v(\emptyset) = \log(1 + \eta L_{ii}).$$

Since $\mathbf{L} \succeq 0$ and $\eta > 0$, $\mathbf{K}_S$ is positive definite and the Schur complement satisfies $s_i > 0$.

Complexity. Exact computation over m positive-reward rollouts requires evaluating all 2^m coalitions. With direct log-determinant evaluation, this costs $O(2^m m^3)$ time in the worst case, plus $O(m2^m)$ time for Shapley aggregation.

With the permutation estimator, each sampled permutation processes the m rollouts once. At step t, the triangular solve and Cholesky update cost $O(t^2)$, so one full permutation costs

$$\sum_{t=1}^{m} O(t^2) = O(m^3).$$

The total complexity is therefore $O(Km^3)$ time and $O(m^2)$ memory. When $K \ll 2^m$, this provides a substantial reduction in computation while preserving an unbiased estimate of the Shapley credits.

Concentration. The estimator also has a high-probability error bound. Under binary verifier rewards, for any $i \in P$, Theorem 2 gives

$$0 \le v(S \cup \{i\}) - v(S) \le b, \qquad b = \log(1 + \eta).$$

Thus $\widehat{\phi}_i$ is the average of K independent bounded random variables. By Hoeffding's inequality,

$$\Pr\left(|\widehat{\phi}_i - \phi_i| \ge \epsilon\right) \le 2 \exp\left(-\frac{2K\epsilon^2}{b^2}\right).$$

Applying a union bound over all $i \in P$ yields

$$\Pr\left(\max_{i\in P} |\widehat{\phi}_i - \phi_i| \ge \epsilon\right) \le 2m \exp\left(-\frac{2K\epsilon^2}{b^2}\right).$$

Equivalently, choosing

$$K \ge \frac{b^2}{2\epsilon^2} \log \frac{2m}{\delta}$$

ensures

$$\max_{i\in P} |\widehat{\phi}_i - \phi_i| \le \epsilon$$

with probability at least $1 - \delta$.

Reward conservation. Although the estimator approximates individual credits, it preserves the team value exactly at the level of each sampled permutation. For any permutation π,

$$\sum_{i \in P} [v(S_i^\pi \cup \{i\}) - v(S_i^\pi)] = v(P) - v(\emptyset) = v(P),$$

because the marginal contributions telescope along the permutation. Averaging over the K sampled permutations gives $\sum_{i \in P} \widehat{\phi}_i = v(P)$. Therefore, using $\widehat{\phi}_i$ in reward redistribution preserves the original batch reward:

$$\widetilde{r}_i = R_{\text{batch}} \frac{\widehat{\phi}_i}{\sum_{j \in P} \widehat{\phi}_j}, \qquad i \in P,$$

and $\widetilde{r}_i = 0$ for $i \notin P$. Hence, $\sum_{i=1}^{G} \widetilde{r}_i = R_{\text{batch}}$, which means the Monte Carlo estimator does not introduce additional reward mass.

B Limitations

Although GCPO improves both reasoning accuracy and solution diversity in our experiments, several limitations remain. First, GCPO requires additional computation to estimate group-level coverage and cooperative credits. In our experiments, this overhead is limited because the rollout group size is moderate and only verifier-correct responses are used for credit assignment. The runtime analysis shows that GCPO remains comparable to existing diversity-aware RLVR baselines while providing stronger empirical gains. Future work will explore approximate credit estimation and more efficient representation strategies to further reduce the training overhead. Second, GCPO relies on semantic embeddings to estimate the diversity and redundancy of reasoning paths. The determinantal team value assumes that distances or angles in the embedding space can meaningfully reflect differences in reasoning strategies. While our ablation studies suggest that GCPO is relatively robust across several sentence encoders, embedding-based similarity remains an imperfect proxy for reasoning-level diversity. Future work may incorporate task-adaptive encoders, structured reasoning representations, or verifier-aware similarity measures to better capture meaningful diversity. A more principled and adaptive diversity measure could better capture underlying semantic variation and automatically adjust to task-specific requirements, as different reasoning tasks may emphasize different aspects of diversity.

C Broader Impacts

Our cooperative policy optimization framework for LLM reasoning offers potential benefits for applications that require diverse and reliable problem solving. By encouraging multiple non-redundant valid reasoning paths, the method can support AI tutoring systems with alternative explanations and solution strategies, and may assist scientific research by facilitating hypothesis generation and exploratory problem solving. It may also improve the robustness of reasoning-oriented LLMs by reducing dependence on a narrow set of repeated solution patterns. However, diverse reasoning does not by itself guarantee factual correctness or safety. When applied to high-stakes domains, such systems should be used with human oversight, domain-specific validation, and appropriate safeguards against biased or incorrect reasoning.

D Detailed Metrics Definition

This section provides formal definitions for the evaluation metrics used throughout the paper. We cover both quality metrics (Pass@k) and diversity metrics (self-BLEU, self-ROUGE, Distinct-n, and LLM-as-Judge).

D.1 Pass@k

The pass@k metric evaluates the probability of obtaining at least one correct response among k independently sampled solutions. While pass@1 measures single-attempt accuracy, pass@k emphasizes the model's ability to generate diverse correct solutions and improve success rates

through sampling. Given a question x and k i.i.d. responses with binary rewards $r_i \in \{0, 1\}$, the pass@k metric is defined as:

$$\text{pass@k} = \mathbb{P}\left[\bigvee_{i=1}^{k}(r_i = 1)\right] = \mathbb{E}\left[1 - \prod_{i=1}^{k}(1 - r_i)\right]. \tag{12}$$

In practice, when we sample exactly $n \geq k$ solutions per problem, we estimate pass@k using the unbiased estimator:

$$\text{pass@k} \approx 1 - \frac{\binom{n-c}{k}}{\binom{n}{k}}, \tag{13}$$

where c is the number of correct solutions among the n samples. This formulation accounts for sampling without replacement and provides a consistent estimate as n increases.

D.2 Self-BLEU

Self-BLEU measures the average n-gram overlap among generated responses, quantifying lexical redundancy. Given a set of N responses $\{o_1, \ldots, o_N\}$ for a single prompt, we compute the BLEU score of each response o_i against all other responses $\{o_j\}_{j \neq i}$ as references. The self-BLEU score is then:

$$\text{self-BLEU} = \frac{1}{N}\sum_{i=1}^{N}\text{BLEU}(o_i, \{o_j\}_{j \neq i}). \tag{14}$$

Higher self-BLEU indicates greater lexical similarity (lower diversity). In our experiments, we report **diversity** as $1 -$ self-BLEU, so that higher values indicate more diverse outputs. We use BLEU-4 with smoothing for all experiments.

D.3 Self-ROUGE

Self-ROUGE measures the average longest common subsequence (LCS) overlap among generated responses. Similar to self-BLEU, for a set of N responses $\{o_1, \ldots, o_N\}$, we compute:

$$\text{self-ROUGE-L} = \frac{1}{N}\sum_{i=1}^{N}\text{ROUGE-L}(o_i, \{o_j\}_{j \neq i}), \tag{15}$$

where ROUGE-L measures the F-score of the longest common subsequence between o_i and the reference set. Higher self-ROUGE indicates greater structural similarity (lower diversity). We report **diversity** as $1 -$ self-ROUGE-L, so that higher values indicate more diverse reasoning structures.

D.4 Distinct-n

Distinct-n measures the ratio of unique n-grams to total n-grams in the generated text, capturing token-level diversity. For a set of responses $\{o_1, \ldots, o_N\}$, let $\mathcal{G}_n$ denote the multiset of all n-grams extracted from these responses. Then:

$$\text{Distinct-n} = \frac{|\text{unique}(\mathcal{G}_n)|}{|\mathcal{G}_n|}, \tag{16}$$

where $|\text{unique}(\mathcal{G}_n)|$ is the number of distinct n-grams and $|\mathcal{G}_n|$ is the total count. Higher Distinct-n indicates greater token-level variety. In our experiments, we primarily report Distinct-1, which captures vocabulary richness across generated solutions.

D.5 LLM-as-Judge Diversity

LLM-as-Judge diversity uses a large language model to evaluate the semantic and reasoning-level diversity of generated solutions. We use GPT5.4 [33] to evaluate the diversity. Given a problem and a set of N responses, we prompt a judge model to score diversity across three dimensions: logic diversity (differences in reasoning flow), formula diversity (differences in mathematical techniques), and semantic diversity (differences in overall approach). Each dimension is scored on a 0–10 scale, and the overall diversity score is the average of the three. The prompt template is shown below.

Prompt Template for LLM-as-Judge Diversity Scoring

System: You are a strict but fair grader. You will be given one math problem and **32 independent rollouts (solutions)**. Your task is to score diversity across rollouts.

Evaluate the following three dimensions on a **0–10 scale**:

- **logic_diversity**: differences in reasoning flow or step order
- **formula_diversity**: differences in formulas, identities, or techniques used
- **semantic_diversity**: differences in overall semantic approach or framing

Then output **overall diversity** as the average of the three scores (**0–10**, rounded to **one decimal**).

Be objective and do not prefer any model. **Output JSON only.**

This metric captures high-level reasoning diversity that lexical metrics (self-BLEU, self-ROUGE, Distinct-n) may miss, providing a complementary view of solution variety.

E Additional Results

E.1 Pass@k Performance

To comprehensively demonstrate GCPO's consistent advantage across different sampling budgets, we evaluate pass@k performance for $k \in \{2, 4, 8, 16, 32, 64\}$ on five mathematical reasoning benchmarks: AIME2024, AIME2025, MATH500, MINERVA, and Olympiad. We compare GCPO against the base model, GRPO, and DQO across two model scales: Qwen3-1.7B and Qwen3-4B.

As shown in Tables 6 and 7, GCPO consistently outperforms all baseline methods across the entire spectrum of pass@k metrics. The performance gap becomes particularly pronounced at higher k values, where GCPO's superior exploration capabilities enable it to discover more diverse correct reasoning paths. This trend holds consistently across all benchmarks and both model scales, confirming that GCPO's cooperative credit assignment mechanism effectively expands the coverage of valid reasoning strategies without sacrificing correctness.

E.2 More Ablation Studies

The team value in Eq. (3) depends on two design choices that are orthogonal to the cooperative credit assignment mechanism: (i) the frozen semantic encoder f_ϕ used to extract $z_i = f_\phi(o_i)$, and (ii) the kernel function used to construct the Gram matrix L. To verify that GCPO's gains do not hinge on a particular embedding model or similarity measure, we ablate each factor in turn. All experiments in this subsection are conducted on Qwen3-4B evaluated on AIME25, keeping the default setting $G = 8, \eta = 1$ so that the reported numbers are directly comparable to the default GCPO row in Table 2 and Table 10.

Semantic extractor. Table 8 compares the default lightweight encoder all-MiniLM-L6-v2 [27] (22M parameters, 384-dim) against four stronger general-purpose sentence encoders: all-mpnet-base-v2 [27], bge-base-en-v1.5 [40], e5-base-v2 [36], and sentence-t5-base [25] (each $\sim$110M parameters, 768-dim). Although the heavier encoders produce richer semantic representations, we observe only minor fluctuations across all four metrics, and no encoder consistently dominates the others. The all-MiniLM-L6-v2 default and the much larger bge-base-en-v1.5 are statistically indistinguishable within the four-metric grid, suggesting that GCPO's exploration gains arise primarily from the **geometric** structure of the log-determinant team value rather than from fine-grained linguistic features of the encoder. Intuitively, once rollouts are reward-gated by r_i and projected onto the unit sphere via $\bar{z}_i = z_i / \|z_i\|_2$, the information that drives Shapley credit is the **relative angles** among correct rollouts, which are preserved across competent sentence encoders. This robustness is desirable in practice: it allows practitioners to choose the cheapest encoder that still produces discriminative directions, keeping the per-step overhead of GCPO feasible relative to the rollout cost.

Table 6: Pass@k performance (%) for Qwen3-1.7B across different benchmarks.

Pass@k	Method	AIME2024	AIME2025	MATH500	MINERVA	Olympiad	Avg.
$k = 2$	Base	16.3	11.4	62.5	20.1	30.8	28.2
	GRPO	24.6	23.9	68.2	25.7	36.0	35.7
	DQO	28.1	26.3	69.4	26.6	37.5	37.6
	GCPO	**40.7**	**30.8**	**70.3**	**26.9**	**39.2**	**41.6**
$k = 4$	Base	20.9	15.7	67.6	22.5	35.1	32.4
	GRPO	30.2	28.3	71.8	28.6	39.4	39.7
	DQO	33.5	31.1	72.7	29.3	40.8	41.5
	GCPO	**49.4**	**38.2**	**73.5**	**30.0**	**41.6**	**46.5**
$k = 8$	Base	25.2	20.5	71.1	25.4	38.3	36.1
	GRPO	37.4	30.8	74.0	31.5	41.1	43.0
	DQO	40.6	34.2	74.8	31.9	42.4	44.8
	GCPO	**56.3**	**44.7**	**75.6**	**32.8**	**43.9**	**50.7**
$k = 16$	Base	30.0	24.8	73.9	27.6	41.0	39.5
	GRPO	43.7	34.0	75.4	33.2	42.8	45.8
	DQO	47.2	37.7	76.3	34.1	43.7	47.8
	GCPO	**59.1**	**52.5**	**77.0**	**34.6**	**45.2**	**53.7**
$k = 32$	Base	34.5	31.4	75.8	29.7	42.3	42.7
	GRPO	49.6	38.5	77.3	35.1	43.6	48.8
	DQO	52.8	42.9	78.0	35.8	44.1	50.7
	GCPO	**62.2**	**59.8**	**78.4**	**36.4**	**46.5**	**56.7**
$k = 64$	Base	39.1	37.5	76.9	32.0	43.8	45.9
	GRPO	54.2	42.4	77.6	36.6	44.3	51.0
	DQO	57.3	46.8	78.2	37.0	45.4	52.9
	GCPO	**63.0**	**64.7**	**79.3**	**37.5**	**47.1**	**58.3**

Kernel function. The default GCPO kernel couples reward and semantic direction through an inner product, $L_{ij} = r_i r_j \langle \bar{z}_i, \bar{z}_j \rangle$, which admits the D-optimal interpretation underlying the volume objective. To test whether this specific form is necessary, we replace the inner product with three alternative similarity measures while keeping the reward gating $r_i r_j$ unchanged, so that incorrect rollouts remain at zero contribution. Concretely, we consider:

- **Dot product**: $L_{ij} = r_i r_j \, \langle \bar{z}_i, \bar{z}_j \rangle$.
- **Gaussian (RBF)**: $L_{ij} = r_i r_j \, \exp\big(-\|\bar{z}_i - \bar{z}_j\|_2^2/(2\sigma^2)\big)$ with $\sigma = 1$.
- **Laplacian**: $L_{ij} = r_i r_j \, \exp(-\|\bar{z}_i - \bar{z}_j\|_1/\sigma)$ with $\sigma = 1$.
- **Polynomial**: $L_{ij} = r_i r_j \, (\langle \bar{z}_i, \bar{z}_j \rangle + c)^d$ with $c = 0.5$, $d = 2$.

Since all alternatives are positive semi-definite, the resulting regularized log-determinant $\log\det(I + \eta L)$ remains a valid team value and the Shapley decomposition in Proposition 1 continues to hold; only the numerical values of the L_{ij} entries change.

Table 9 reports the results. The default dot-product kernel attains the best Pass@1 and Avg@16, while the Gaussian kernel is marginally better on Pass@16 and LLM-as-Judge diversity, likely because its distance-based saturation further attenuates near-duplicate pairs. The polynomial kernel slightly boosts diversity at the cost of accuracy, a pattern consistent with its sharper penalization of redundant directions. Across all variants, every GCPO kernel substantially exceeds the GRPO baseline , corroborating that the improvements attributed to GCPO stem from the cooperative formulation rather than from a particular choice of similarity. We adopt the dot-product kernel in the main experiments because it is the cheapest to evaluate, directly matches the DPP L-ensemble identity, and offers the strongest quality metrics among the candidates.

Taken together, the two ablations in this subsection indicate that GCPO is insensitive to the specific instantiations of f_ϕ and the similarity function, provided that the encoder yields discriminative directions and the kernel is positive semi-definite. This supports our interpretation that the cooperative

Table 7: Pass@k performance (%) for Qwen3-4B across different benchmarks.

Pass@k	Method	AIME2024	AIME2025	MATH500	MINERVA	Olympiad	Avg.
$k = 2$	Base	26.1	21.8	64.3	27.2	36.4	35.2
	GRPO	42.8	41.2	65.1	30.4	41.9	44.3
	DQO	47.2	44.8	66.4	30.7	43.1	46.4
	GCPO	**58.2**	**53.1**	**69.3**	**31.0**	**44.7**	**51.3**
$k = 4$	Base	31.9	27.7	65.8	30.6	39.7	39.1
	GRPO	47.9	46.8	67.2	32.6	43.8	47.7
	DQO	53.6	51.5	68.5	32.9	44.6	50.2
	GCPO	**66.7**	**58.9**	**72.1**	**33.2**	**46.4**	**55.5**
$k = 8$	Base	38.6	34.3	67.5	32.9	42.2	43.1
	GRPO	52.8	51.7	69.2	34.1	45.3	50.6
	DQO	59.3	55.6	71.4	34.4	46.0	53.3
	GCPO	**72.4**	**62.7**	**74.8**	**34.9**	**47.6**	**58.5**
$k = 16$	Base	44.5	41.4	68.3	35.1	44.2	46.7
	GRPO	58.1	56.9	71.6	35.3	46.7	53.7
	DQO	64.4	61.2	73.6	35.7	47.2	56.4
	GCPO	**75.9**	**66.4**	**76.8**	**36.2**	**48.7**	**60.8**
$k = 32$	Base	49.3	47.8	70.2	35.7	46.1	49.8
	GRPO	62.7	59.9	73.4	36.9	47.5	56.1
	DQO	68.2	64.1	75.4	37.3	48.6	58.7
	GCPO	**78.3**	**69.8**	**78.7**	**37.8**	**49.1**	**62.7**
$k = 64$	Base	51.8	52.7	71.2	37.4	46.8	52.0
	GRPO	66.9	63.2	75.7	37.6	48.1	58.3
	DQO	71.7	67.3	77.8	37.7	48.9	60.7
	GCPO	**79.8**	**73.1**	**80.2**	**38.4**	**49.6**	**64.2**

Table 8: Performance comparison across different semantic extractors f_ϕ on Qwen3-4B/AIME25. The default encoder is shaded in gray. GCPO is robust to the choice of encoder, with differences across models remaining within ± 0.5 on Pass@1 and ± 0.08 on LLM-as-Judge.

Encoder	Params	Dim	Quality		Diversity	
			Pass@1 (%)	Avg@16	Pass@16 (%)	LLM-as-Judge
all-MiniLM-L6-v2	22M	384	**49.02**	**45.46**	66.47	2.58
all-mpnet-base-v2	109M	768	48.88	45.30	66.52	2.60
bge-base-en-v1.5	109M	768	48.95	45.42	**66.71**	**2.62**
e5-base-v2	109M	768	48.72	45.18	66.38	2.55
sentence-t5-base	110M	768	48.64	45.08	66.25	2.53

log-determinant team value, together with Shapley-based credit assignment, is the primary driver of GCPO's exploration benefits.

Hyperparameter Sensitivity. Table 10 studies the rollout group size G and the team-value scale η in Eq. (3). All GCPO variants outperform GRPO, indicating that the improvements are not tied to a fragile hyperparameter choice. With $\eta = 1$, increasing G from 4 to 8 improves both quality and diversity, while $G = 16$ brings only marginal diversity gains and slightly lower accuracy, suggesting increasing redundancy among extra rollouts. Fixing $G = 8$, a smaller η weakens the volume signal, whereas a larger η slightly increases diversity but reduces accuracy. These trends support the default setting $G = 8, \eta = 1$, which provides the best balance between accurate reasoning and diverse coverage.

E.3 Case Study

In this section, we present generated samples during testing. One sample with 16 test-time rollouts, GCPO achieves 7 correct solutions, compared to only 2 for GRPO. We show two example responses

Table 9: Performance comparison across different kernel functions on Qwen3-4B/AIME25. All variants share the reward gating $r_i r_j$ and the same default encoder. The default is shaded in gray.

Kernel	Quality		Diversity	
	Pass@1 (%)	Avg@16	Pass@16 (%)	LLM-as-Judge
Dot product	**49.02**	**45.46**	66.47	2.58
Gaussian (RBF, $\sigma = 1$)	48.85	45.24	**66.73**	**2.64**
Laplacian ($\sigma = 1$)	48.40	44.88	66.32	2.56
Polynomial ($c = 0.5, d = 2$)	48.62	45.02	66.55	2.61

Table 10: Performance comparison across different hyperparameters.

Method	Quality		Diversity	
	Pass@1 (%)	Avg@16	Pass@16 (%)	LLM-as-Judge
GRPO($G = 8$)	33.33	35.21	58.13	1.96
$\eta = 1$, $G = 4$	47.30	43.85	63.20	2.37
$\eta = 1$, $G = 8$	**49.02**	**45.46**	**66.47**	2.58
$\eta = 1$, $G = 16$	48.70	45.10	66.36	**2.63**
$\eta = 0.5$, $G = 8$	48.15	44.70	64.80	2.43
$\eta = 1.0$, $G = 8$	**49.02**	**45.46**	66.47	2.58
$\eta = 1.5$, $G = 8$	48.35	44.85	**66.85**	**2.62**

generated by GCPO, which arrive at the correct answer through distinct reasoning processes, illustrating its ability to explore multiple valid solution paths. In contrast, we present two responses generated by GRPO, which exhibit more limited diversity. Moreover, we present solutions with redundant steps generated by DQO. This demonstrates that GCPO improves exploration without sacrificing solution fidelity: by maintaining controlled diversity across rollouts, the model avoids pattern collapse while consistently discovering correct reasoning paths, leading to stronger overall performance.

E.3.1 Samples generated by GCPO

Two generated samples by GCPO are shown below, and the yellow boxes highlight the different solution strategies used to solve this problem.

Generated Sample: Answer 1

Problem.
Four unit squares form a 2×2 grid. Each of the 12 unit line segments forming the sides of the squares is colored either red or blue in such a way that each unit square has 2 red sides and 2 blue sides. Find the number of such colorings. **Ground Truth:** 82

Answer.
We need to count the number of ways to color the 12 edges of a 2×2 grid such that each of the four unit squares has exactly 2 red sides and 2 blue sides.

Step 1: Identify the structure
The 2×2 grid has:

- 8 boundary edges (forming the outer perimeter)
- 4 interior edges (two vertical and two horizontal)

Each unit square shares some edges with adjacent squares, so the coloring of interior edges affects multiple squares simultaneously.

Step 2: Condition on interior edges

Key insight: Use casework on the number of red interior edges.
Let k denote the number of red interior edges among the four interior edges. We classify all colorings by $k \in \{0, 1, 2, 3, 4\}$.

For each value of k, we determine:

1. How many ways to choose which interior edges are red
2. For each such choice, how many valid boundary colorings exist

Step 3: Case analysis

Case $k = 0$: All interior edges are blue.
Each square has 0 red edges from interior, so both boundary edges of each square must be red. This uniquely determines all boundary edges.
Count: $\binom{4}{0} \times 1 = 1$.

Case $k = 4$: All interior edges are red.
Each square has 2 red edges from interior, so both boundary edges of each square must be blue. This uniquely determines all boundary edges.
Count: $\binom{4}{4} \times 1 = 1$.

Case $k = 1$: Exactly one interior edge is red.
There are $\binom{4}{1} = 4$ ways to choose which interior edge is red.
For each choice, the two squares adjacent to that red interior edge each need exactly 1 more red edge (from their boundary edges), while the other two squares need 2 red boundary edges each.

By carefully tracking the boundary constraints and propagating the forced choices around the perimeter, we find that each interior choice yields exactly 4 valid boundary colorings.

Count: $4 \times 4 = 16$.

Case $k = 3$: Exactly three interior edges are red.
By symmetry (swapping red $\leftrightarrow$ blue), this case has the same count as $k = 1$.
Count: 16.

Case $k = 2$: Exactly two interior edges are red.
There are $\binom{4}{2} = 6$ ways to choose which two interior edges are red. These split into two subcases:
Subcase (i): Opposite interior edges. The two red interior edges are opposite (both vertical or both horizontal). There are 2 such patterns.
For each pattern, systematic enumeration shows 16 valid boundary colorings.
Contribution: $2 \times 16 = 32$.
Subcase (ii): Adjacent interior edges. The two red interior edges form an L-shape. There are 4 such patterns.
For each pattern, systematic enumeration shows 4 valid boundary colorings.
Contribution: $4 \times 4 = 16$.
Total for $k = 2$: $32 + 16 = 48$.

Step 4: Sum all cases

$$1 + 16 + 48 + 16 + 1 = \boxed{82}.$$

Final Answer

$$\boxed{82}$$

Generated Sample: Answer 2

Problem.
Four unit squares form a 2×2 grid. Each of the 12 unit line segments forming the sides of the squares is colored either red or blue in such a way that each unit square has 2 red sides and 2 blue sides. Find the number of such colorings. **Ground Truth:** 82

Answer.
We count the valid colorings by a column-by-column dynamic programming argument. The key is to condition on the colors of the two vertical interior edges that separate the left and right columns.

Step 1: Define the interface state

Key insight: DP over boundary/interface states.
Instead of classifying colorings by the total number of red interior edges, we sweep the grid from left to right and use the two vertical interior edges as the interface state:

$$s = (v_t, v_b) \in \{00, 01, 10, 11\},$$

where v_t is the upper vertical interior edge, v_b is the lower vertical interior edge, and 1 denotes red while 0 denotes blue.

For each fixed interface state s, we count the number of ways to complete the left column and the right column.

Step 2: Count valid completions of the left column
For a fixed interface state $s = (v_t, v_b)$, let $L(s)$ be the number of valid colorings of the left column. The left column contains one horizontal interior edge, denoted by

$$h_\ell \in \{0, 1\}.$$

Given $s = (v_t, v_b)$ and h_ℓ, the top-left square already has $v_t + h_\ell$ red edges from its two shared/interior edges. Therefore, it needs

$$2 - v_t - h_\ell$$

red edges among its two outer boundary edges.
Similarly, the bottom-left square needs

$$2 - v_b - h_\ell$$

red edges among its two outer boundary edges.
Hence,

$$L(s) = \sum_{h_\ell \in \{0,1\}} \binom{2}{2 - v_t - h_\ell} \binom{2}{2 - v_b - h_\ell},$$

where a binomial coefficient is treated as zero if its lower index is outside $\{0, 1, 2\}$.

Step 3: Evaluate the four interface states
We compute $L(s)$ for each possible state.

$$L(00) = \binom{2}{2}\binom{2}{2} + \binom{2}{1}\binom{2}{1} = 1 + 4 = 5,$$
$$L(01) = \binom{2}{2}\binom{2}{1} + \binom{2}{1}\binom{2}{0} = 2 + 2 = 4,$$
$$L(10) = \binom{2}{1}\binom{2}{2} + \binom{2}{0}\binom{2}{1} = 2 + 2 = 4,$$
$$L(11) = \binom{2}{1}\binom{2}{1} + \binom{2}{0}\binom{2}{0} = 4 + 1 = 5.$$

Thus,

$$L(00) = 5, \quad L(01) = 4, \quad L(10) = 4, \quad L(11) = 5.$$

Step 4: Combine the two columns

By symmetry, the number of valid completions of the right column with the same interface state is

$$R(s) = L(s).$$

Once the vertical interface state s is fixed, the left and right column completions are independent. Therefore, the total number of valid colorings is obtained by summing over all interface states:

$$\sum_{s \in \{00,01,10,11\}} L(s)R(s) = L(00)^2 + L(01)^2 + L(10)^2 + L(11)^2.$$

Substituting the values above gives

$$5^2 + 4^2 + 4^2 + 5^2 = 25 + 16 + 16 + 25 = 82.$$

Final Answer

$$\boxed{82}$$

E.3.2 Samples generated by GRPO

Two samples generated by GRPO are shown below, and the red boxes highlight the erroneous solution strategies that lead to failure.

Generated Sample: Answer 1

Problem.
Four unit squares form a 2×2 grid. Each of the 12 unit line segments forming the sides of the squares is colored either red or blue in such a way that each unit square has 2 red sides and 2 blue sides. Find the number of such colorings. **Ground Truth:** $\boxed{82}$

Answer.
We need to count the number of ways to color the 12 edges of a 2×2 grid such that each unit square has exactly 2 red sides and 2 blue sides.

Step 1: Count edges and constraints

The grid has 12 edges total. Each of the 4 unit squares imposes the constraint that exactly 2 of its 4 edges are red.

Step 2: Count colorings per square

Logical error starts here (incorrect independence assumption).
Consider each unit square independently. For each square, we need to choose 2 edges out of 4 to be red. This gives $\binom{4}{2} = 6$ choices per square.
Since there are 4 squares, the total number of colorings would be:

$$6^4 = 1296.$$

Step 3: Account for shared edges

However, adjacent squares share edges. There are 4 interior edges that are each shared by 2 squares.
When we count 6^4, we are overcounting because we're treating shared edges as if they can be colored independently for each square.

Step 4: Apply correction factor

Each interior edge is counted twice in our 6^4 calculation (once for each adjacent square). We need to divide by the number of ways these 4 interior edges can be "double-counted."
Since each interior edge has 2 possible colors, and we've counted each twice, we apply a correction:

$$\text{Corrected count} = \frac{6^4}{2^4} = \frac{1296}{16} = 81.$$

Wait, this doesn't seem right. Let me reconsider.
Actually, the correction should account for the fact that 4 edges are shared. A better approach: we have 8 boundary edges (not shared) and 4 interior edges (shared).
For the 4 interior edges, once we fix their colors, the boundary edges are partially constrained.
Let's estimate:

$$\text{Total} \approx \frac{6^4}{4} = \frac{1296}{4} = 324.$$

Final Answer

$$\boxed{324}$$

Generated Sample: Answer 2

Problem.
Four unit squares form a 2×2 grid. Each of the 12 unit line segments forming the sides of the squares is colored either red or blue in such a way that each unit square has 2 red sides and 2 blue sides. Find the number of such colorings. **Ground Truth:** $\boxed{82}$

Answer.
We are asked to count colorings of a 2×2 grid where each unit square has exactly 2 red sides and 2 blue sides.

Step 1: Count total red edges

Logical error starts here (incorrect counting argument).
Since each of the 4 squares has exactly 2 red edges, counting red edges by summing over all squares gives:

$$\text{Total red edge count} = 4 \times 2 = 8.$$

Let r_i be the number of red interior edges and r_b be the number of red boundary edges. Then:

$$2r_i + r_b = 8.$$

Since there are 4 interior edges and 8 boundary edges, we have:

$$r_i \in \{0, 1, 2, 3, 4\}, \quad r_b = 8 - 2r_i.$$

For r_b to be valid (i.e., $0 \le r_b \le 8$), we need $0 \le r_i \le 4$, which is always satisfied.

Step 2: Count colorings for each case

For each valid value of r_i, the number of colorings is:

$$\binom{4}{r_i} \times \binom{8}{r_b} = \binom{4}{r_i} \times \binom{8}{8 - 2r_i}.$$

Here, $\binom{4}{r_i}$ counts the ways to choose which interior edges are red, and $\binom{8}{8-2r_i}$ counts the ways to choose which boundary edges are red.

Step 3: Sum over all cases
Summing over $r_i \in \{0, 1, 2, 3, 4\}$:

$$\begin{aligned}\text{Total} &= \sum_{r_i=0}^{4} \binom{4}{r_i} \times \binom{8}{8-2r_i} \\ &= \binom{4}{0}\binom{8}{8} + \binom{4}{1}\binom{8}{6} + \binom{4}{2}\binom{8}{4} + \binom{4}{3}\binom{8}{2} + \binom{4}{4}\binom{8}{0} \\ &= 1 \cdot 1 + 4 \cdot 28 + 6 \cdot 70 + 4 \cdot 28 + 1 \cdot 1 \\ &= 1 + 112 + 420 + 112 + 1 \\ &= 646.\end{aligned}$$

Final Answer

$$\boxed{646}$$

E.3.3 Sample generated by DQO with redundant steps

One sample generated by DQO is shown below. While it arrives at the correct answer using the same core strategy as GCPO Answer 1, the gray boxes highlight superficial reformulations that introduce alternative expressions or intermediate steps without contributing genuinely new solution insights.

Generated Sample: DQO Answer

Problem.
Four unit squares form a 2×2 grid. Each of the 12 unit line segments forming the sides of the squares is colored either red or blue in such a way that each unit square has 2 red sides and 2 blue sides. Find the number of such colorings. **Ground Truth:** $\boxed{82}$

Answer.
We need to count the number of ways to color the 12 edges of a 2×2 grid such that each of the four unit squares has exactly 2 red sides and 2 blue sides.

Step 1: Identify the structure
The 2×2 grid has 8 boundary edges and 4 interior edges (two vertical and two horizontal). Each unit square shares some edges with adjacent squares, so the coloring of interior edges affects multiple squares simultaneously.

Step 2: Condition on interior edges
Key insight: Use casework on the number of red interior edges.
Let k denote the number of red interior edges among the four interior edges. We classify all colorings by $k \in \{0, 1, 2, 3, 4\}$.
For each value of k, we determine how many ways to choose which interior edges are red, and for each such choice, how many valid boundary colorings exist.

Step 3: Case analysis
Case $k = 0$: All interior edges are blue.
Each square has 0 red edges from interior, so both boundary edges of each square must be red. This uniquely determines all boundary edges.

Superficial reformulation: The number of ways to select 0 interior edges to be red is $\binom{4}{0} = \frac{4!}{0!\cdot 4!} = 1$. For this unique selection, there is exactly 1 valid boundary coloring.

Count: $1 \times 1 = 1$.

Case $k = 4$: All interior edges are red.
Each square has 2 red edges from interior, so both boundary edges of each square must be blue. This uniquely determines all boundary edges.

Superficial reformulation: The number of ways to select 4 interior edges to be red is $\binom{4}{4} = \frac{4!}{4!\cdot 0!} = 1$. For this unique selection, there is exactly 1 valid boundary coloring.

Count: $1 \times 1 = 1$.

Case $k = 1$: Exactly one interior edge is red.

Superficial reformulation: The number of ways to select 1 interior edge to be red is $\binom{4}{1} = \frac{4!}{1!\cdot 3!} = \frac{4\cdot 3!}{1\cdot 3!} = 4$.

For each choice, the two squares adjacent to that red interior edge each need exactly 1 more red edge (from their boundary edges), while the other two squares need 2 red boundary edges each.
By carefully tracking the boundary constraints and propagating the forced choices around the perimeter, we find that each interior choice yields exactly 4 valid boundary colorings.
Count: $4 \times 4 = 16$.

Case $k = 3$: Exactly three interior edges are red.

Superficial reformulation: By the red-blue symmetry (swapping red $\leftrightarrow$ blue), colorings with $k = 3$ red interior edges correspond bijectively to colorings with $4 - 3 = 1$ red interior edges. Therefore, this case has the same count as $k = 1$.

Count: 16.

Case $k = 2$: Exactly two interior edges are red.

Superficial reformulation: The number of ways to select 2 interior edges to be red is $\binom{4}{2} = \frac{4!}{2!\cdot 2!} = \frac{4\cdot 3}{2\cdot 1} = 6$.

These 6 ways split into two geometric types:
Subcase (i): Opposite interior edges. The two red interior edges are opposite (both vertical or both horizontal).

Superficial reformulation: There are 2 such patterns: selecting both vertical interior edges gives $\binom{2}{2} = 1$ way, and selecting both horizontal interior edges gives $\binom{2}{2} = 1$ way. Total: $1 + 1 = 2$ patterns.

For each pattern, systematic enumeration shows 16 valid boundary colorings.
Contribution: $2 \times 16 = 32$.
Subcase (ii): Adjacent interior edges. The two red interior edges form an L-shape.

Superficial reformulation: We can choose 1 vertical interior edge from 2 options and 1 horizontal interior edge from 2 options, giving $\binom{2}{1} \times \binom{2}{1} = 2 \times 2 = 4$ patterns.

For each pattern, systematic enumeration shows 4 valid boundary colorings.
Contribution: $4 \times 4 = 16$.
Total for $k = 2$: $32 + 16 = 48$.

Step 4: Sum all cases

Superficial reformulation: Summing over all cases:

$$\text{Total} = 1 + 16 + 48 + 16 + 1 = (1 + 1) + (16 + 16) + 48 = 2 + 32 + 48 = 82.$$

$$1 + 16 + 48 + 16 + 1 = \boxed{82}.$$

Final Answer

$$\boxed{82}$$

F Theoretical Analysis of GCPO

We provide complete proofs for the theoretical statements in Section 3.2. Throughout this section, for each rollout $i \in [G]$, let

$$z_i = f_\phi(o_i), \qquad \bar{z}_i = \frac{z_i}{\|z_i\|_2}, \qquad \tilde{z}_i = r_i \bar{z}_i, \qquad r_i \in \{0, 1\}.$$

For a coalition $S \subseteq [G]$, let $\tilde{Z}_S \in \mathbb{R}^{|S| \times d}$ be the matrix whose rows are $\{\tilde{z}_i\}_{i \in S}$. Then the principal kernel matrix satisfies

$$L_S = \tilde{Z}_S \tilde{Z}_S^\top,$$

and the team value is

$$v(S) = \log \det(I_{|S|} + \eta L_S), \qquad \eta > 0.$$

We use $v(\emptyset) = 0$.

F.1 Proof of Proposition 1

Proof. Since L_S is a Gram matrix, it is symmetric positive semi-definite. Thus it admits an eigendecomposition

$$L_S = U \operatorname{diag}(\lambda_1, \ldots, \lambda_{|S|}) U^\top,$$

where U is orthogonal and $\lambda_\ell \geq 0$ for all ℓ. Therefore,

$$I_{|S|} + \eta L_S = U \operatorname{diag}(1 + \eta\lambda_1, \ldots, 1 + \eta\lambda_{|S|}) U^\top.$$

Taking determinants gives

$$\det(I_{|S|} + \eta L_S) = \prod_{\ell=1}^{|S|} (1 + \eta\lambda_\ell),$$

and hence

$$v(S) = \log \det(I_{|S|} + \eta L_S) = \sum_{\ell=1}^{|S|} \log(1 + \eta\lambda_\ell).$$

We now verify the three consequences.

First, suppose $r_i = 0$. Then $\tilde{z}_i = 0$. For any coalition $S \subseteq [G] \setminus \{i\}$, after ordering i last, we have

$$L_{S \cup \{i\}} = \begin{bmatrix} L_S & 0 \\ 0^\top & 0 \end{bmatrix}.$$

Therefore,

$$I + \eta L_{S \cup \{i\}} = \begin{bmatrix} I + \eta L_S & 0 \\ 0^\top & 1 \end{bmatrix},$$

so

$$\det(I + \eta L_{S \cup \{i\}}) = \det(I + \eta L_S).$$

Thus $v(S \cup \{i\}) = v(S)$, meaning that any zero-reward rollout is a null player.

Second, suppose S contains m verifier-correct rollouts with mutually orthogonal normalized embeddings. Since $r_i = 1$ for all $i \in S$ and $\langle \bar{z}_i, \bar{z}_j \rangle = \mathbf{1}\{i = j\}$, we have

$$L_S = I_m.$$

Thus all m eigenvalues are equal to 1, and

$$v(S) = \sum_{\ell=1}^{m} \log(1+\eta) = m \log(1+\eta).$$

Third, suppose S contains m verifier-correct rollouts with identical normalized embeddings $\bar{z}_i = \bar{z}$ for all $i \in S$. Then

$$L_S = \mathbf{1}_m \mathbf{1}_m^\top,$$

whose eigenvalues are $m, 0, \ldots, 0$. Hence

$$v(S) = \log(1 + \eta m).$$

This proves the proposition. □

F.2 Proof of Theorem 2

Proof. By definition,

$$v(S) = \log\det(I_{|S|} + \eta \tilde{Z}_S \tilde{Z}_S^\top).$$

Using Sylvester's determinant identity,

$$\det(I_{|S|} + \eta \tilde{Z}_S \tilde{Z}_S^\top) = \det(I_d + \eta \tilde{Z}_S^\top \tilde{Z}_S).$$

Therefore,

$$v(S) = \log\det(I_d + \eta \tilde{Z}_S^\top \tilde{Z}_S).$$

Adding rollout i to S appends the row $\tilde{z}_i^\top$ to $\tilde{Z}_S$. Hence

$$\tilde{Z}_{S\cup\{i\}}^\top \tilde{Z}_{S\cup\{i\}} = \tilde{Z}_S^\top \tilde{Z}_S + \tilde{z}_i \tilde{z}_i^\top.$$

Since $\tilde{z}_i = r_i \bar{z}_i$, this becomes

$$\tilde{Z}_{S\cup\{i\}}^\top \tilde{Z}_{S\cup\{i\}} = \tilde{Z}_S^\top \tilde{Z}_S + r_i^2 \bar{z}_i \bar{z}_i^\top.$$

Let

$$A_S = I_d + \eta \tilde{Z}_S^\top \tilde{Z}_S.$$

Because $\eta > 0$ and $\tilde{Z}_S^\top \tilde{Z}_S \succeq 0$, we have $A_S \succ 0$. Then

$$\Delta_i(S) = \log\det\left(A_S + \eta r_i^2 \bar{z}_i \bar{z}_i^\top\right) - \log\det(A_S).$$

By the matrix determinant lemma,

$$\det(A_S + \eta r_i^2 \bar{z}_i \bar{z}_i^\top) = \det(A_S)\left(1 + \eta r_i^2 \bar{z}_i^\top A_S^{-1} \bar{z}_i\right).$$

Substituting this into the expression for $\Delta_i(S)$ yields

$$\Delta_i(S) = \log\left(1 + \eta r_i^2 \bar{z}_i^\top A_S^{-1} \bar{z}_i\right),$$

which is the desired formula.

It remains to prove the bound. Since $A_S = I_d + \eta \tilde{Z}_S^\top \tilde{Z}_S \succeq I_d$, we have

$$0 \preceq A_S^{-1} \preceq I_d.$$

Because $\|\bar{z}_i\|_2 = 1$, it follows that

$$0 \le \bar{z}_i^\top A_S^{-1} \bar{z}_i \le \bar{z}_i^\top \bar{z}_i = 1.$$

Therefore,

$$1 \le 1 + \eta r_i^2 \bar{z}_i^\top A_S^{-1} \bar{z}_i \le 1 + \eta r_i^2.$$

Taking logarithms gives

$$0 \le \Delta_i(S) \le \log(1 + \eta r_i^2).$$

This completes the proof. □

F.3 Proof of Theorem 3

Proof. For any subset $S \subseteq C$ with $|S| = s$, all rows of $\tilde{Z}_S$ are equal to $u^\top$. Thus

$$L_S = \tilde{Z}_S \tilde{Z}_S^\top = \|u\|_2^2 \mathbf{1}_s \mathbf{1}_s^\top.$$

The matrix $\mathbf{1}_s \mathbf{1}_s^\top$ has one nonzero eigenvalue s and $s-1$ zero eigenvalues. Therefore, L_S has one nonzero eigenvalue $s\|u\|_2^2$, and

$$v(S) = \log(1 + \eta s \|u\|_2^2).$$

In particular,

$$v(C) = \log(1 + \eta m \|u\|_2^2).$$

The rollouts in C are symmetric: for any two $i, j \in C$ and any $S \subseteq C \setminus \{i, j\}$,

$$v(S \cup \{i\}) = v(S \cup \{j\}).$$

Therefore, by the symmetry property of the Shapley value, all rollouts in C receive the same Shapley credit:

$$\phi_1 = \cdots = \phi_m.$$

By the efficiency property of the Shapley value,

$$\sum_{i=1}^{m} \phi_i = v(C) - v(\emptyset) = v(C),$$

where $v(\emptyset) = 0$. Hence

$$m\phi_i = \log(1 + \eta m \|u\|_2^2),$$

and thus

$$\phi_i = \frac{1}{m} \log(1 + \eta m \|u\|_2^2).$$

This proves the theorem. □

Comparison with leave-one-out. For the same isolated duplicate group C, the leave-one-out score of any $i \in C$ is

$$\begin{aligned} \mathrm{LOO}_i &= v(C) - v(C \setminus \{i\}) \\ &= \log(1 + \eta m \|u\|_2^2) - \log(1 + \eta(m-1)\|u\|_2^2) \\ &= \log \frac{1 + \eta m \|u\|_2^2}{1 + \eta(m-1)\|u\|_2^2}. \end{aligned}$$

Let $c = \eta\|u\|_2^2 > 0$ and define $f(s) = \log(1 + cs)$. Since f is concave and $f(0) = 0$, its average slope satisfies

$$\frac{f(m) - f(0)}{m} \geq f(m) - f(m-1).$$

Therefore,

$$\frac{1}{m} \log(1 + cm) \geq \log \frac{1 + cm}{1 + c(m-1)}.$$

That is,

$$\phi_i \geq \mathrm{LOO}_i.$$

This shows that leave-one-out can underestimate the shared value of duplicated discoveries, whereas Shapley fairly distributes the shared value among all rollouts realizing the same reasoning path.

F.4 Proof of Proposition 4

Proof. By reward conservation, for every batch B,

$$\frac{1}{G}\sum_{i=1}^{G} R_{\mathrm{GCPO}}(\tau_i \mid B) = \frac{1}{G}\sum_{i=1}^{G} R_{\mathrm{obj}}(\tau_i).$$

Taking expectation over $B \sim \pi^G$ gives

$$\begin{aligned} J^{\Sigma}_{\mathrm{GCPO}}(\pi) &= \mathbb{E}_{B\sim\pi^G}\left[\frac{1}{G}\sum_{i=1}^{G} R_{\mathrm{obj}}(\tau_i)\right] \\ &= \frac{1}{G}\sum_{i=1}^{G} \mathbb{E}_{\tau_i\sim\pi}\left[R_{\mathrm{obj}}(\tau_i)\right] \\ &= \mathbb{E}_{\tau\sim\pi}\left[R_{\mathrm{obj}}(\tau)\right] \\ &= J_{\mathrm{obj}}(\pi). \end{aligned}$$

Thus the two scalar objectives are equal pointwise for every policy π, so they have the same set of maximizers. If the same regularizer $\Omega(\pi)$ is added to both objectives, then

$$J^{\Sigma}_{\mathrm{GCPO}}(\pi) - \Omega(\pi) = J_{\mathrm{obj}}(\pi) - \Omega(\pi)$$

for every π, so the maximizers remain identical. □

Remark. This proposition establishes invariance of the batch-sum scalar objective under reward conservation. It does not imply that every per-rollout credit-assignment policy-gradient estimator is identical to the original verifier-reward gradient. GCPO intentionally changes the per-rollout credit assignment within a batch; Proposition 5 characterizes this routing effect.

F.5 Proof of Proposition 5

Proof. First, since $\mu_{\tilde{r}} = \frac{1}{G}\sum_{i=1}^{G} \tilde{r}_i$, we have

$$\sum_{i=1}^{G}(\tilde{r}_i - \mu_{\tilde{r}}) = \sum_{i=1}^{G} \tilde{r}_i - G\mu_{\tilde{r}} = 0.$$

Therefore,

$$\sum_{i=1}^{G} A_i^{\mathrm{GCPO}} = \frac{1}{\sigma_{\tilde{r}} + \epsilon_{\mathrm{num}}}\sum_{i=1}^{G}(\tilde{r}_i - \mu_{\tilde{r}}) = 0.$$

Similarly,

$$\sum_{i=1}^{G} A_i^{\mathrm{GRPO}} = 0.$$

Thus,

$$\sum_{i=1}^{G} \Delta A_i = \sum_{i=1}^{G}\left(A_i^{\mathrm{GCPO}} - A_i^{\mathrm{GRPO}}\right) = 0.$$

Now consider the unclipped policy-gradient surrogate

$$\mathcal{L}_{\mathrm{GCPO}}(\theta) = \mathbb{E}_{x\sim\mathcal{D}}\left[\frac{1}{G}\sum_{i=1}^{G} \log \pi_\theta(o_i \mid x) A_i^{\mathrm{GCPO}}\right],$$

where A_i^{GCPO} is treated as a stop-gradient quantity. Then

$$\nabla_\theta \mathcal{L}_{\mathrm{GCPO}} = \mathbb{E}_{x\sim\mathcal{D}}\left[\frac{1}{G}\sum_{i=1}^{G} \nabla_\theta \log \pi_\theta(o_i \mid x) A_i^{\mathrm{GCPO}}\right].$$

Using

$$A_i^{\mathrm{GCPO}} = A_i^{\mathrm{GRPO}} + \Delta A_i,$$

we obtain

$$\begin{aligned}
\nabla_\theta \mathcal{L}_{\mathrm{GCPO}} &= \mathbb{E}_{x\sim\mathcal{D}}\left[\frac{1}{G}\sum_{i=1}^{G} \nabla_\theta \log \pi_\theta(o_i \mid x)\left(A_i^{\mathrm{GRPO}} + \Delta A_i\right)\right] \\
&= \mathbb{E}_{x\sim\mathcal{D}}\left[\frac{1}{G}\sum_{i=1}^{G} \nabla_\theta \log \pi_\theta(o_i \mid x) A_i^{\mathrm{GRPO}}\right] + \mathbb{E}_{x\sim\mathcal{D}}\left[\frac{1}{G}\sum_{i=1}^{G} \nabla_\theta \log \pi_\theta(o_i \mid x)\Delta A_i\right] \\
&= \nabla_\theta \mathcal{L}_{\mathrm{GRPO}} + \mathbb{E}_{x\sim\mathcal{D}}\left[\frac{1}{G}\sum_{i=1}^{G} \nabla_\theta \log \pi_\theta(o_i \mid x)\Delta A_i\right].
\end{aligned}$$

This proves the decomposition. □

Remark on PPO clipping. The above decomposition is exact for the unclipped policy-gradient surrogate with stop-gradient advantages. For the PPO-style clipped objective and KL penalty used in implementation, the same algebraic decomposition should be interpreted as a first-order explanation of the routing signal rather than an exact equality of the full clipped gradients.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: Yes, the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Please refer to the Conclusion Section and Appendix.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: The paper has some theoretical analysis. We provide a complete and correct analysis in Appendix.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Please refer to Experiment Section and Appendix.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: The datasets used are publicly available, the code will be released at https://github.com/bradybuddiemarch/gcpo.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Please refer to Experiment Section and Appendix.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: Please refer to Experiment Section.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Please refer to the implementation details in the Appendix

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research conducted in the paper conform with the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We list potential positive societal impacts in the Appendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We properly credit data, paper, and ideas that we used in this paper.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: We document well about the asset.

    Guidelines:

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

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

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

    Answer: [N/A]

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

    Guidelines:

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

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

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

    Answer: [N/A]

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

    Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [Yes]

Justification: We decrible the pipeline to post-training the LLM in the Experiment Section.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.