PODA: PARTIAL-ORDER ADVANTAGE ESTIMATION FOR MULTI-REWARD LLM ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-reward reinforcement learning for large language models usually compresses every reward vector into a scalar advantage with a fixed linear rule. This is convenient, but it inserts pairwise preferences even when two rollouts trade off different objectives. We introduce PODA, a prompt-local advantage estimator that instead constructs a dominance graph, combines an inverse dominance-count score with an antisymmetric dominance margin, and leaves the PPO/GRPO policy objective unchanged. PODA is a credit-assignment rule applied after rewards and any task priorities have been specified; it does not infer user preferences or learn a Pareto coverage set. Across mathematical reasoning and tool calling with two and three rewards, and across Qwen2.5, Qwen3, and Qwen3.5 models, PODA improves both primary performance and constraint satisfaction over equally tuned linear, lexicographic, rank-sum, additive-indicator, hypervolume, GRPO, GDPO, and DAPO comparators. On the representative two-reward settings, it improves the strongest matched indicator baseline by 0.91 math-accuracy points and 0.55 tool-accuracy points; the corresponding three-reward gains are 0.9 and 0.8 points. Paired three-seed confidence intervals are positive in the representative settings, while the estimator adds about 1% step time at group size 16. Training and dominance diagnostics show the largest gains when reward conflict and graph sparsity make fixed scalarization or count-only scoring least informative.

1 INTRODUCTION

Modern language-model training rarely optimizes a single notion of quality. Reasoning systems balance correctness against computation, tool agents must complete tasks while obeying an interface, and safety training combines several behavioral criteria (Christiano et al., 2017; Ziegler et al., 2019; Ouyang et al., 2022; Dai et al., 2023; Lai et al., 2024; Shao et al., 2024; Guo et al., 2025; Qian et al., 2025; Yu et al., 2025; Liu et al., 2026). The reward vector can express these criteria, yet the policy update still needs one scalar advantage per rollout. How that scalar is constructed is therefore not bookkeeping; it determines which within-group comparisons receive credit.

GRPO-family pipelines commonly sum rewards before group normalization. GDPO improves this step by normalizing each reward dimension separately, but then still forms a fixed linear aggregate (Liu et al., 2026). A weighted utility is appropriate when the weights encode the desired preference. It is less defensible as a default credit rule when the task specification provides several valid criteria but no reason to compare every conflicting pair. A mathematically correct but longer response and a shorter incorrect response, for example, should not acquire a direct preference merely because one fixed projection happens to rank them. We call the resulting, projection-dependent pairwise distinctions *scalarization bias*.

The aim here is deliberately narrower than preference learning or multi-objective policy-set construction. The reward channels—and any explicit priority such as lexicographic correctness—are fixed before our estimator is applied. We ask how a prompt-local group of reward vectors should be converted into credit once that task specification is in place. Pareto dominance supplies only comparisons justified by all active objectives: a rollout receives a directed edge when it is no worse on every objective and better on at least one. Incomparable pairs receive no direct edge, although their final scalar graph scores may differ through relations with other rollouts. Figure 1 isolates this

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

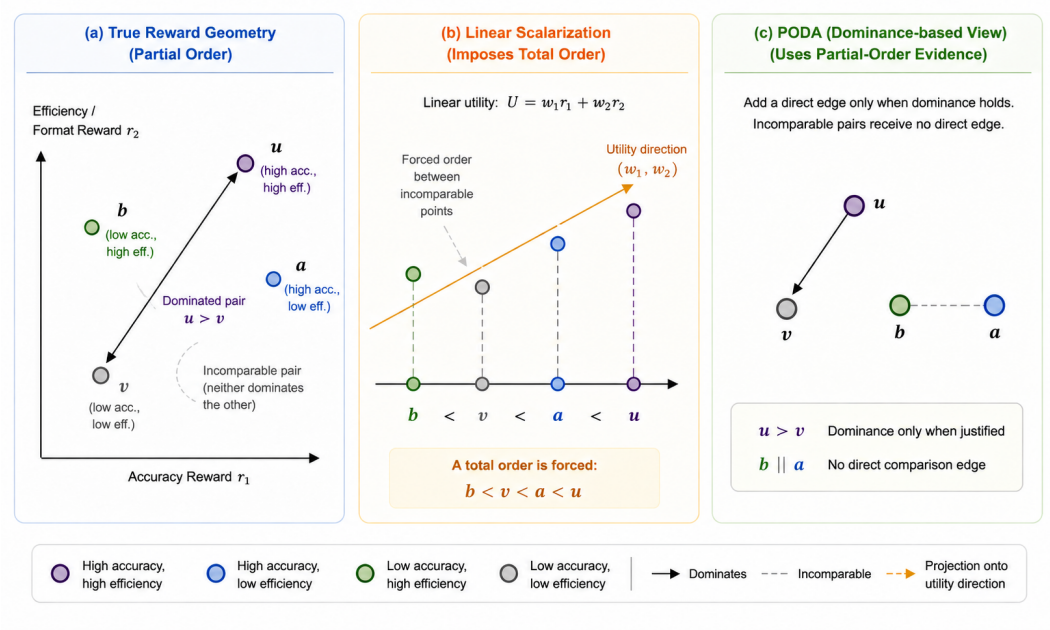

Figure 1: Credit assignment from multi-reward outcomes. Linear scalarization directly orders every pair along a fixed utility direction. PODA inserts an edge only for a Pareto-dominating pair. An incomparable pair receives no direct edge, although its two rollouts may still obtain different scalar scores through their relations with other group members.

distinction: the estimator uses partial-order evidence without claiming to preserve incomparability after scalar scoring.

We instantiate this credit rule in PODA (**P**artial-**O**rder **D**ominance **A**dvantage Estimation). PODA builds a prompt-local dominance graph, scores graph position with an inverse incoming-edge count, and adds a signed margin flow that retains the strength of observed dominance. The score is whitened exactly where GRPO expects an advantage, so rollout generation, clipping, KL control, and the reference policy are unchanged. For a fixed behavior policy, the resulting score-function estimator optimizes the expected nonlinear graph utility assigned to one rollout against sampled peers. It makes no claim of globally improving hypervolume or recovering an unknown preference.

We compare against equally tuned fixed linear weights, lexicographic order, Borda/rank-sum, additive ϵ-indicator and hypervolume-contribution estimators, as well as GRPO, GDPO, and DAPO. The evaluation covers two- and three-reward variants of mathematical reasoning and tool calling, five model/task settings across several Qwen scales, and a matched Qwen3.5 transfer. Three-seed confidence intervals, convergence curves, reward-correlation trajectories, graph-density diagnostics, and wall-clock measurements test whether the gains are stable and whether they follow the mechanism predicted by the estimator.

Our contributions are as follows.

- We isolate the reward-vector-to-advantage map as a distinct design choice and characterize when fixed scalarization inserts unsupported pairwise preferences.
- We propose a low-overhead graph-derived estimator that combines an inverse dominance count and signed margin flow while retaining the standard PPO/GRPO objective.
- We show consistent gains under matched tuning across $K = 2$ and $K = 3$ rewards, multiple Qwen generations and scales, and strong indicator- and recipe-level baselines, with positive paired confidence intervals on representative settings and approximately 1% step-time overhead.

2 Background and Related Work

2.1 Scalar advantages in LLM reinforcement learning

RLHF and direct or online preference optimization ultimately train with scalar feedback (Christiano et al., 2017; Ziegler et al., 2019; Stiennon et al., 2020; Ouyang et al., 2022; Rafailov et al., 2023; Schulman et al., 2017). GRPO-style reasoning systems retain this interface while replacing learned critics with group-relative scores (Shao et al., 2024; Guo et al., 2025; Hu, 2025; Zheng et al., 2025; Yu et al., 2025; Liu et al., 2025c). GDPO is the closest normalization-centered method: it standardizes reward dimensions separately before summation, reducing reward collapse without changing the linear ordering rule (Liu et al., 2026). PODA acts at the same interface but replaces the final sum with a graph score.

2.2 Multi-objective alignment and indicator methods

Multi-reward alignment appears in safety, personalization, reasoning, and structured behavior (Dai et al., 2023; Mu et al., 2024; Jang et al., 2023; Lai et al., 2024). Objective-level approaches learn Pareto-stationary updates or adapt reward weights with gradient and hypervolume information (Sener & Koltun, 2018; He & Maghsudi, 2025; Li et al., 2025; Lu et al., 2025); preference-conditioned and MORL methods instead seek a family or coverage set of policies (Roijers et al., 2013; Hayes et al., 2022; Pirotta et al., 2015; Nguyen et al., 2025). Those methods solve a broader problem than prompt-local credit assignment and are not interchangeable with a drop-in group estimator. The closest indicator-based alternative is IB-GRPO, which derives group-relative advantages from an additive ϵ-indicator for learning-path recommendation (Wang et al., 2026). Our matched comparison includes this indicator family and hypervolume contribution. PODA differs by combining dominance-graph topology with signed edge strength, then testing the same estimator across reasoning and tool use.

2.3 Reward conflict in reasoning and tool use

Length-aware reasoning methods explicitly expose the tension between correctness and computation (Luo et al., 2025b; Aggarwal & Welleck, 2025; Liu et al., 2025a; Luo et al., 2025a; Liu et al., 2025b; Arora & Zanette, 2025; Yi & Wang, 2025; Su & Cardie, 2025; Cui et al., 2025). Tool-use systems expose a different interaction: structural validity can conflict with task completion on some samples while supporting it on others (Qian et al., 2025; Liu et al., 2024; Lin et al., 2024; Zhang et al., 2025; Patil et al., 2025). These two domains let us test both conflicting and partly synergistic reward relationships. Classical dominance, indicator, and evolutionary methods establish the relevant partial-order tools (Deb et al., 2002; Roijers et al., 2013; Hayes et al., 2022); our focus is the narrower question of converting a sampled rollout group into one scalar advantage per response.

3 Method

3.1 Problem setup and scope

For a prompt x, the policy samples a group of G candidate rollouts $\{y_j\}_{j=1}^{G}$. Each rollout y_j receives a K-dimensional reward vector

$$\mathbf{r}_j = [r_{j,1}, \ldots, r_{j,K}] \in \mathbb{R}^K,$$

where $r_{j,k}$ is the reward of rollout j on objective k. Existing GRPO-style estimators collapse this vector to a scalar score of the form

$$s_j^{\text{lin}} = \sum_{k=1}^{K} w_k \phi_k(r_{j,k}), \tag{1}$$

where w_k is a fixed weight for objective k and $\phi_k(\cdot)$ is either the identity map or a per-dimension normalization operator. This setup isolates the design choice studied here: the reward functions, rollout policy, PPO clipping, and KL regularization remain unchanged, and only the map from a reward vector to a scalar advantage is modified. The reward channels encode the task specification. If one objective is categorically primary, a lexicographic constraint or feasibility mask is applied before graph construction; PODA does not infer that priority from data.

3.2 From scalarization to a prompt-local graph

If rollout u is better than rollout v on one objective but worse on another, the pair is incomparable under Pareto dominance even though a linear utility forces one above the other. This forced distinction depends on the chosen projection, not on unanimous reward evidence. PODA therefore begins from a directed graph: strict dominance creates an edge, while an incomparable pair creates none. The graph is prompt-local and policy-dependent because it is constructed from the rollouts sampled together.

3.3 Partial-Order Dominance Advantage Estimation

Step 1: groupwise normalization. Let μ_k and σ_k denote the mean and standard deviation of reward dimension k within the current prompt group, and let $\epsilon > 0$ be a small numerical constant. When decoupled normalization is enabled, PODA first computes

$$a_{j,k} = \frac{r_{j,k} - \mu_k}{\sigma_k + \epsilon}, \tag{2}$$

and otherwise uses $a_{j,k} = r_{j,k}$. We write $\mathbf{a}_j = [a_{j,1}, \ldots, a_{j,K}]$ for the resulting normalized reward vector of rollout j.

Step 2: dominance relation. Let $\tau \geq 0$ be a tolerance parameter that suppresses numerically insignificant differences. Rollout u is said to dominate rollout v if

$$\mathbf{a}_u \succ \mathbf{a}_v \iff (\forall k,\ a_{u,k} \geq a_{v,k} - \tau) \wedge (\exists k,\ a_{u,k} > a_{v,k} + \tau)\,. \tag{3}$$

This definition requires u to be no worse than v on every objective up to tolerance τ, and strictly better on at least one objective.

Step 3: inverse dominance-count score. For rollout j, count its incoming dominance edges:

$$\mathrm{PR}_j = 1 + \sum_{m \neq j} \mathbb{I}(\mathbf{a}_m \succ \mathbf{a}_j), \tag{4}$$

where $\mathbb{I}(\cdot)$ is the indicator function. The associated inverse-count score is then

$$s_j^{\mathrm{count}} = \frac{1}{\mathrm{PR}_j}. \tag{5}$$

Non-dominated rollouts receive $\mathrm{PR}_j = 1$ and therefore the largest count score. This quantity is not recursive non-dominated-sorting depth: it distinguishes rollouts by the number of peers that dominate them.

Step 4: dominance margin. The inverse count score captures front structure but can be too coarse when many rollouts share the same incoming-edge count. To recover discrimination among these ties, PODA measures the strength of their incoming and outgoing dominance relations. For a dominating pair (u, v), define the per-objective gap

$$\Delta_{u,v,k} = a_{u,k} - a_{v,k},$$

and let $m(u, v)$ denote a scalar margin:

$$m(u,v) = \begin{cases} \frac{1}{K}\sum_k \Delta_{u,v,k}, & \text{mean}, \\ \min_k \Delta_{u,v,k}, & \text{min}, \\ \alpha \min_k \Delta_{u,v,k} + (1-\alpha)\frac{1}{K}\sum_k \Delta_{u,v,k}, & \text{mixed}. \end{cases} \tag{6}$$

Here $\alpha \in [0, 1]$ controls the interpolation between a mean-style and worst-dimension notion of dominance strength. The net margin of rollout j is

$$M_j = \sum_{m \neq j} \mathbb{I}(\mathbf{a}_j \succ \mathbf{a}_m) m(j,m) - \sum_{m \neq j} \mathbb{I}(\mathbf{a}_m \succ \mathbf{a}_j) m(m,j). \tag{7}$$

The first term accumulates outgoing-edge strength and the second subtracts incoming-edge strength. This antisymmetric construction makes the margin groupwise zero-sum. Because the tolerance can admit a small negative coordinate gap, $m(u, v)$ is intentionally signed. The inverse count score supplies the edge direction, while λ is calibrated so the margin refines rather than arbitrarily overrides that ordering; Proposition 1 in Appendix F states the exact pairwise condition.

Algorithm 1 PODA within one prompt group

Require: Reward vectors $\{\mathbf{r}_j\}_{j=1}^{G}$, tolerance τ, margin weight λ, count/margin switches
1: Optionally normalize each reward dimension within the group to obtain $\{\mathbf{a}_j\}_{j=1}^{G}$
2: Build the dominance graph using Eq. (3): add $u \to v$ whenever $\mathbf{a}_u \succ \mathbf{a}_v$
3: **for** $j = 1, \ldots, G$ **do**
4: Compute $\mathrm{PR}_j = 1 + \sum_{m \neq j} \mathbb{I}(\mathbf{a}_m \succ \mathbf{a}_j)$ and $s_j^{\mathrm{count}} = 1/\mathrm{PR}_j$
5: Aggregate the net dominance margin M_j from all incoming and outgoing dominance edges
6: Form $s_j = \mathbb{I}_{\mathrm{count}}/\mathrm{PR}_j + \mathbb{I}_{\mathrm{margin}}\lambda M_j$
7: Whiten $\{s_j\}_{j=1}^{G}$ within the group and pass the resulting advantages to the unchanged PPO/GRPO loss

Step 5: scalar advantage. Let $\mathbb{I}_{\mathrm{count}}, \mathbb{I}_{\mathrm{margin}} \in \{0, 1\}$ indicate whether the count term and the margin term are active, and let $\lambda \geq 0$ weight the margin contribution. The pre-whitened scalar score is

$$s_j = \mathbb{I}_{\mathrm{count}} \frac{1}{\mathrm{PR}_j} + \mathbb{I}_{\mathrm{margin}} \lambda M_j, \tag{8}$$

which is optionally whitened within the prompt group:

$$A_j = \frac{s_j - \mu_s}{\sigma_s + \epsilon}. \tag{9}$$

Here $\mu_s = \frac{1}{G}\sum_{j=1}^{G} s_j$ and σ_s is the corresponding within-group standard deviation. The final scalar advantage A_j is the only quantity passed to the unchanged PPO/GRPO loss.

Practical calibration of τ and λ. The two estimator-specific hyperparameters play different roles. The tolerance τ controls when nearby reward values should be treated as effectively tied, and therefore primarily changes the density of the dominance graph. The margin weight λ controls the relative scale of dense local discrimination versus the coarse incoming-count signal. In the normalized reward space used here, stable configurations fall in a relatively small range: Appendix H shows that $\tau \in [0.01, 0.05]$ and $\lambda \in [0.1, 0.4]$ produce the most consistent results across both tasks. A practical heuristic is to choose the smallest τ that suppresses obvious reward jitter within a group, then tune λ so that the typical magnitude of λM_j is comparable to that of $1/\mathrm{PR}_j$ early in training.

Intuition. Each component of PODA has a distinct role. Groupwise normalization removes scale imbalance across reward dimensions. Dominance comparison avoids inserting a direct edge for an incomparable pair. The inverse count score favors rollouts with fewer incoming edges, while the margin term restores local discrimination when many rollouts have the same count. The result is neither a weighted sum nor a bare edge count; it is a scalar graph signal that uses partial-order evidence while remaining compatible with group-relative optimization.

3.4 Optimization interpretation and estimator relationships

Let $\bar{\pi}$ be the behavior policy that samples the other $G - 1$ rollouts and let $\Psi_{\mathrm{PODA}}(y, \mathcal{Y}_-)$ be the scalar graph score assigned to rollout y in that group. For fixed $\bar{\pi}$, PODA induces the nonlinear scalar objective

$$J_{\mathrm{PODA}}(\theta; \bar{\pi}) = \mathbb{E}_x \mathbb{E}_{y \sim \pi_\theta(\cdot|x)} \mathbb{E}_{\mathcal{Y}_- \sim \bar{\pi}^{G-1}(\cdot|x)} \left[\Psi_{\mathrm{PODA}}(y, \mathcal{Y}_-) \right]. \tag{10}$$

At $\pi_\theta = \bar{\pi}$, the score-function gradient of Eq. (10) is the expected group-gradient estimator used by PODA; PPO clipping supplies the usual local surrogate. Thus the method optimizes an explicit policy-dependent graph utility. It does not guarantee global Pareto or hypervolume improvement, and it still outputs one scalar advantage.

Dominance and task preference play different roles. Reward design decides which outcomes matter, and an explicit rule such as lexicographic correctness decides whether one criterion is categorically primary. PODA is applied only after those choices. It then uses dominance as a minimal consistency condition for credit assignment: if one sampled rollout is no worse on every active reward and better on at least one, the graph records that comparison. When two rollouts conflict, the graph abstains from adding a direct edge instead of resolving the conflict with an arbitrary projection. This does not make their final scalar scores equal; other incoming and outgoing edges can still separate them.

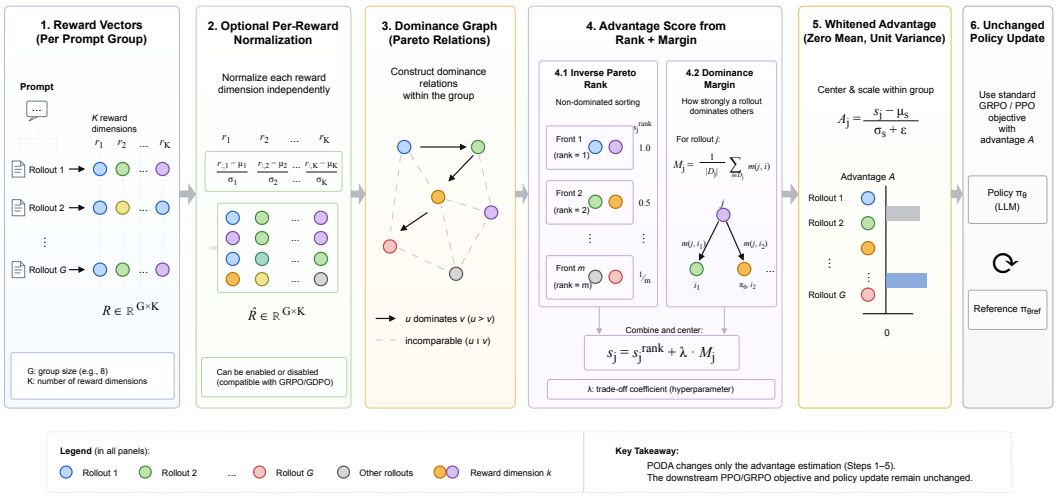

Figure 2: Overview of PODA. Only the reward-vector-to-advantage map changes; rollout generation, clipping, KL control, and the policy objective remain unchanged.

The graph utility is therefore not a substitute for preference-conditioned multi-objective learning. It is a nonlinear, group-dependent scalar surrogate for training one policy under a specified reward system. The matched lexicographic and nine-weight linear controls in Section 4 test whether the observed gains can be recovered simply by declaring a primary objective or tuning a fixed projection. They cannot in the studied settings: PODA improves the primary metric and the constraint together in both the conflicting math rewards and the partly synergistic tool rewards.

GRPO aggregates reward channels before group normalization, whereas GDPO normalizes each channel before a linear sum. PODA may use the same per-channel normalization as GDPO, then replaces the linear sum with the graph score. Setting $\lambda = 0$ yields count-only PODA, not GRPO or GDPO; disabling the count term yields margin-only PODA. Formal properties, including order preservation on strict two-rollout edges, count-score collapse under sparse dominance, the zero-sum margin, and the empty-graph boundary, appear in Appendix F. Complexity is $O(G^2K)$, with measured overhead reported in Section 4.5.

4 Experiments

4.1 Experimental setup

Tasks, models, and metrics. Mathematical reasoning uses the DeepScaleR preview data and its matched GDPO-style protocol with Qwen3-1.7B, Qwen3-4B, and Qwen3-8B (Luo et al., 2025b; Liu et al., 2026; Yang et al., 2025). We report Pass@1 accuracy on MATH, AIME, AMC, Minerva-style subsets, and OlympiadBench, together with Exceed, the percentage of generations beyond the protocol length limit (Hendrycks et al., 2021; Lewkowycz et al., 2022; Mathematical Association of America, 2024a;b; He et al., 2024). Tool calling uses the BFCL-v3 protocol with Qwen2.5-Instruct-1.5B and 3B; metrics are Live, Multi-turn, Non-live, and average accuracy plus correct-format rate (Qian et al., 2025; Patil et al., 2025; Qwen Team et al., 2025). The two-reward settings use correctness and length compliance for math, and correctness and format validity for tools. The three-reward settings add answer-format validity for math and tool efficiency for tool calling.

Matched comparison. All estimator comparisons share reward vectors, normalization, rollouts, checkpoints, training steps, seeds, and hyperparameter-search budget. Math uses $G = 16$ and tool calling uses $G = 8$. The baselines include GRPO, GRPO without standard-deviation scaling, GDPO, DAPO, the best of nine fixed linear weights, lexicographic order, Borda/rank-sum, additive ϵ-indicator, and hypervolume contribution. These estimator-level controls are directly comparable; PAMA, GAPO, and dynamic weighting change objective-level gradients or weights and are therefore

Table 1: Two-reward mathematical reasoning across Qwen3 scales (Acc ↑ / Exc ↓, %). Under the matched protocol, PODA improves accuracy over GDPO by 2.38, 1.96, and 1.86 points on average while further reducing Exceed. Bold marks the best result in each model block.

		Qwen3-1.7B			Qwen3-4B			Qwen3-8B		
		GRPO	GDPO	PODA	GRPO	GDPO	PODA	GRPO	GDPO	PODA
MATH	Acc ↑	90.3	90.9	**92.6**	93.4	93.4	**94.3**	94.2	94.5	**95.4**
	Exceed ↓	1.4	0.3	**0.2**	0.8	**0.1**	**0.1**	0.6	**0.1**	**0.1**
AIME	Acc ↑	42.7	44.2	**46.9**	54.1	56.4	**58.7**	60.3	62.2	**64.8**
	Exceed ↓	1.9	0.4	**0.3**	2.5	**0.1**	**0.1**	1.2	0.2	**0.1**
AMC	Acc ↑	77.9	78.6	**81.8**	84.7	83.8	**86.2**	86.6	87.0	**88.5**
	Exceed ↓	0.9	**0.2**	**0.2**	0.7	**0.1**	**0.1**	0.5	**0.1**	**0.1**
Minerva	Acc ↑	46.5	46.9	**49.1**	51.9	51.4	**53.6**	54.3	54.6	**56.7**
	Exceed ↓	0.5	**0.1**	**0.1**	0.3	**0.1**	**0.1**	0.3	**0.1**	**0.1**
Olympiad	Acc ↑	56.4	57.3	**59.5**	66.3	67.0	**69.0**	70.1	70.9	**73.1**
	Exceed ↓	3.3	0.9	**0.6**	1.6	1.0	**0.7**	1.6	0.5	**0.4**
Average	Acc ↑	62.80	63.60	**65.98**	70.08	70.40	**72.36**	73.10	73.84	**75.70**
Average	Exceed ↓	1.60	0.42	**0.28**	1.18	0.28	**0.22**	0.84	0.20	**0.16**

Table 2: Two-reward BFCL-v3 tool calling (%). PODA improves GDPO average accuracy by 0.99 and 1.23 points for the 1.5B and 3B models while also increasing format validity.

	Qwen2.5-Instruct-1.5B			Qwen2.5-Instruct-3B		
	GRPO	GDPO	PODA	GRPO	GDPO	PODA
Live Acc ↑	50.63	55.36	**56.80**	69.23	71.22	**72.60**
Multi-turn Acc ↑	2.04	2.50	**2.90**	3.14	4.59	**5.10**
Non-live Acc ↑	37.87	40.58	**41.70**	45.24	46.79	**48.00**
Avg Acc ↑	30.18	32.81	**33.80**	39.20	40.87	**42.10**
Correct Format ↑	76.33	80.66	**82.00**	81.64	82.23	**84.10**

positioned in related work rather than inserted into an unmatched table. Unless stated otherwise, non-public results are means over three seeds. Appendix D gives implementation details, and Appendix E gives uncertainty and tuning sweeps.

The absolute math numbers should be read as within-protocol comparisons, not as replacements for results obtained with different prompts, sampling rules, or decoding budgets. This restriction does not weaken the estimator comparison because every method in a table uses the same protocol.

4.2 Main two-reward results

Tables 1 and 2 show gains in all five model/task settings rather than in one checkpoint. The gains are pronounced on AIME and Olympiad, where correctness and the length constraint conflict sharply. On BFCL-v3, the same estimator improves task completion and valid structure together, showing that PODA does not require every reward pair to be negatively correlated. Table 3 further shows that the representative gains remain positive across three matched seeds.

4.3 Strong estimators, three rewards, and model transfer

The comparison in Table 4 gives every estimator the same inputs and search budget. The best non-PODA result is the additive ϵ-indicator on math and DAPO on tool accuracy. PODA improves the strongest indicator-style estimator by 0.91 math-accuracy points and 0.55 tool-accuracy points, while satisfying both constraints. The margin over DAPO on tools is smaller (0.20 accuracy and 0.10 format), so the evidence supports a consistent estimator gain rather than a uniformly large advantage over every training recipe.

The recipe-level control reaches the same conclusion. Replacing DAPO's default group score with PODA raises math accuracy from 70.74 to 71.46 while reducing Exceed from 0.33 to 0.17; on tools, accuracy rises from 41.90 to 42.80 and correct format from 84.00 to 84.90. The credit-assignment gain

Table 3: Three-seed uncertainty on representative settings. Intervals are paired 95% confidence intervals for the PODA–GDPO difference; PODA wins all three seeds.

Setting	GDPO	PODA	Paired 95% CI
Qwen3-4B AIME	56.4 ± 0.4	$\mathbf{58.7 \pm 0.3}$	$[+1.25, +3.35]$
Qwen3-4B math avg.	70.40 ± 0.12	$\mathbf{72.36 \pm 0.10}$	$[+1.10, +2.82]$
Qwen3-4B Exceed	0.28 ± 0.03	$\mathbf{0.22 \pm 0.02}$	lower in 3/3
Qwen2.5-3B tool avg.	40.87 ± 0.18	$\mathbf{42.10 \pm 0.16}$	$[+0.72, +1.74]$
Qwen2.5-3B format	82.23 ± 0.15	$\mathbf{84.10 \pm 0.14}$	$[+1.15, +2.59]$

Table 4: Equally tuned estimators on Qwen3-4B math and Qwen2.5-3B tools. Math reports average accuracy / Exceed; tools report average accuracy / correct format.

Estimator	Math ↑ / ↓	Tool ↑ / ↑
Best fixed linear (9 weights)	70.90 / 0.26	41.70 / 83.60
Lexicographic	71.05 / 0.24	41.35 / 83.55
Borda/rank-sum	71.20 / 0.25	41.45 / 83.60
Additive ϵ-indicator	71.45 / 0.23	41.55 / 83.70
Hypervolume contribution	71.35 / 0.23	41.50 / 83.65
DAPO	70.74 / 0.33	41.90 / 84.00
PODA	**72.36 / 0.22**	**42.10 / 84.10**

therefore survives a change in the surrounding optimization recipe. We treat this as a compatibility result, not a new optimizer claim; the complete component-by-recipe table is in Appendix G.

Table 5 shows that increasing the reward dimension makes the partial-order test harder: the $K = 3$ incomparable-pair rate is 42% for math and 30% for tools, with empty-graph rates of 22% and 18%. Despite this sparsity, PODA improves the best indicator by 0.9 and 0.8 accuracy points and by 3.3 and 3.9 normalized joint-success points. The Qwen3.5 transfer adds a model-family check: accuracy rises from 69.4 to 70.3 without worsening the secondary constraint. Figure 3 confirms that the gain emerges during training rather than only at the final checkpoint. These results establish the tested scope as $K = 2$–3 and Qwen2.5/Qwen3/Qwen3.5, not arbitrary reward dimension or preference-conditioned MORL.

4.4 Ablations

Count-only improves GDPO, but the full estimator adds 1.66 math-accuracy points and 0.7 tool-accuracy points over the count term alone. Margin-only is weaker, which shows that edge strength cannot replace graph position; its value is complementary. The no-normalization variant remains above GDPO, isolating the ordering rule from decoupled normalization. Appendix Table 10 gives the compact cross-domain comparison; Appendix G retains the complete ablations and their visualization, while Appendix H sweeps λ, τ, and group size.

4.5 Optimization and graph diagnostics

The two tasks expose different reward relationships. Math correctness and length compliance are negatively correlated, with Spearman correlations of $-0.42/-0.31/-0.24$ from early to late training; the incomparable-pair rate rises from 0.38 to 0.47. Tool correctness and format are partly synergistic, with correlations of $+0.28/+0.34/+0.39$, yet incomparability still rises from 0.24 to 0.31. Across groups, PODA gain correlates with incomparable rate at $\rho = 0.46$ for math and 0.34 for tools. These are mechanism-consistent associations, not causal estimates; Appendix Figure 5 shows the underlying reward geometry and conflict-stratified gains.

Group-level diagnostics explain why both score terms are needed. In math, dominance density falls from 0.34 to 0.12 and count-score standard deviation from 0.18 to 0.07, while the full-minus-count performance gap grows from 0.12 to 0.42. Tools show the same pattern. The count signal is informative when the graph is dense; the margin carries more of the discrimination as incoming counts collapse. Empty graphs are the exact zero-signal boundary. They remain uncommon for

Table 5: Additional reward dimension and model transfer. Joint success is normalized to GDPO=100.

Setting	GDPO	Best indicator	PODA
$K = 3$ math avg. accuracy	70.8	71.3	**72.2**
$K = 3$ math joint success	100.0	102.5	**105.8**
$K = 3$ tool avg. accuracy	41.0	41.4	**42.2**
$K = 3$ tool joint success	100.0	102.2	**106.1**
Qwen3.5 matched accuracy	–	69.4	**70.3**

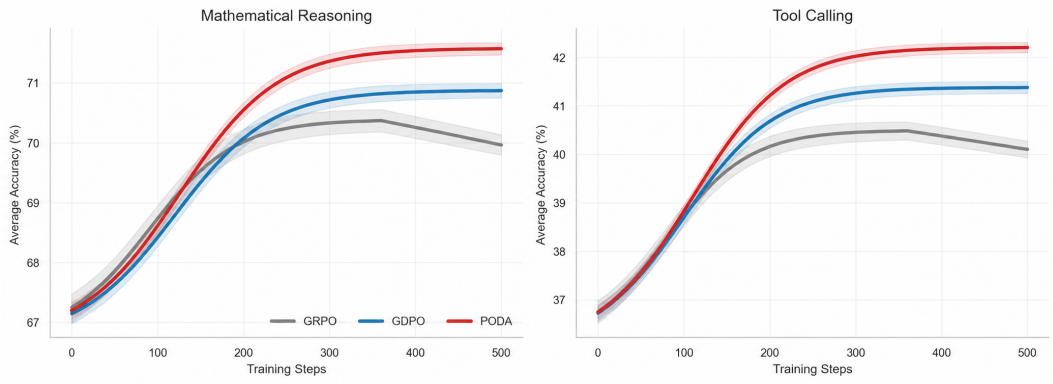

Figure 3: Training trajectories on the representative math and tool settings. Curves report validation average accuracy under the matched two-reward protocols; shaded regions show across-seed variation. PODA separates from GDPO during training rather than only at the final checkpoint; GRPO degrades late in both runs.

$K = 2$ but reach 22%/18% in the tested $K = 3$ settings, directly marking where higher-dimensional extensions will need crowding or other secondary evidence. Full statistics appear in Appendix I.

The graph construction is cheap at the studied group sizes. At $G = 16$, PODA takes $1.01\times$ GDPO step time and 31.5 versus 31.4 GB of peak memory; math/tool training uses 41.4/18.6 versus 41.2/18.5 GPU-hours. Measured overhead is 0.8%, 1.0%, and 1.8% for $G = 8, 16, 24$. Appendix H reports the absolute timings.

5 Limitations

PODA assumes that the supplied reward channels and any hard priority relation are meaningful; it cannot recover a missing user preference. It trains one policy with a nonlinear scalar graph utility, not a Pareto or coverage set. Evidence is limited to $K = 2$–3 and the Qwen2.5/Qwen3/Qwen3.5 model generations. Dominance becomes sparse as K grows, and an empty graph yields zero advantage, so the present estimator should not be extrapolated to high-dimensional reward spaces without an additional tie-break. Gradient-adaptive and dynamic-weighting methods change the objective-level training interface, whereas Table 4 isolates estimators that consume the same sampled reward vectors; our results do not claim superiority over those broader methods. Finally, the $O(G^2K)$ comparison remains negligible for $G \leq 24$ here but will become visible for much larger groups.

6 Conclusion

Multi-reward GRPO must turn reward vectors into scalar credit. PODA does so with a prompt-local dominance graph, inverse incoming-edge count, and signed margin rather than a fixed projection. Across two- and three-reward math and tool settings, it outperforms equally tuned linear, lexicographic, indicator, hypervolume, GDPO, and DAPO controls at about 1% step-time overhead. Gains persist across seeds and grow with reward conflict and graph sparsity. Comparison geometry is therefore an empirically consequential design choice in multi-reward LLM reinforcement learning.

AI USE STATEMENT

Generative AI tools assisted literature search, manuscript structuring, drafting and language editing, LaTeX inspection, figure-text revision, reference formatting, and claim–evidence checks on author-supplied results. They did not generate measurements, run experiments, or replace any reported experiment. The authors manually verified the cited sources and every numerical transcription, reviewed all AI-assisted revisions, and take responsibility for the final paper.

REPRODUCIBILITY STATEMENT

The estimator is fully specified in Section 3 and Algorithm 1. Appendix D reports training hardware, task recipes, group sizes, optimization settings, and estimator hyperparameters; Appendices E–I provide seed variability, matched tuning controls, sensitivity analyses, timing, proofs, and graph diagnostics. These details separate the estimator change from the otherwise unchanged PPO/GRPO pipeline.

REFERENCES

Pranjal Aggarwal and Sean Welleck. L1: Controlling how long a reasoning model thinks with reinforcement learning. *arXiv preprint arXiv:2503.04697*, 2025.

Daman Arora and Andrea Zanette. Training language models to reason efficiently. *arXiv preprint arXiv:2502.04463*, 2025.

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

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

Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. Safe-rlhf: Safe reinforcement learning from human feedback. *arXiv preprint arXiv:2310.12773*, 2023.

Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

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

Conor F. Hayes, Roxana Rădulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa M. Zintgraf, Diederik M. Roijers, Enda Howley, Patrick Mannion, Ann Nowé, Gabriel Ramos, Marcello Restelli, Peter Vamplew, and Shimon Whiteson. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*, 36(1):26, 2022.

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.

Qiang He and Setareh Maghsudi. Pareto multi-objective alignment for language models. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 257–272. Springer, 2025.

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

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

Joel Jang, Seungone Kim, Bill Yuchen Lin, Yizhong Wang, Jack Hessel, Luke S. Zettlemoyer, Hannaneh Hajishirzi, Yejin Choi, and Prithviraj Ammanabrolu. Personalized soups: Personalized large language model alignment via post-hoc parameter merging. *arXiv preprint arXiv:2310.11564*, 2023.

Yuhang Lai, Siyuan Wang, Shujun Liu, Xuanjing Huang, and Zhongyu Wei. Alarm: Align language models via hierarchical rewards modeling. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2024.

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.

Chengao Li, Hanyu Zhang, Yunkun Xu, Hongyan Xue, Xiang Ao, and Qing He. Gradient-adaptive policy optimization: Towards multi-objective alignment of large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11214–11232, 2025. doi: 10.18653/v1/2025.acl-long.549. URL https://aclanthology.org/2025.acl-long.549/.

Qiqiang Lin, Muning Wen, Qiuying Peng, Guanyu Nie, Junwei Liao, Jun Wang, Xiaoyun Mo, Jiamu Zhou, Cheng Cheng, Yin Zhao, et al. Hammer: Robust function-calling for on-device language models via function masking. *arXiv preprint arXiv:2410.04587*, 2024.

Shih-Yang Liu, Xin Dong, Ximing Lu, Shizhe Diao, Mingjie Liu, Min-Hung Chen, Hongxu Yin, Yu-Chiang Frank Wang, Kwang-Ting Cheng, Yejin Choi, Jan Kautz, and Pavlo Molchanov. Dler: Doing length penalty right – incentivizing more intelligence per token via reinforcement learning. *arXiv preprint arXiv:2510.15110*, 2025a.

Shih-Yang Liu, Xin Dong, Ximing Lu, Shizhe Diao, Peter Belcak, Mingjie Liu, Min-Hung Chen, Hongxu Yin, Yu-Chiang Frank Wang, Kwang-Ting Cheng, Yejin Choi, Jan Kautz, and Pavlo Molchanov. Gdpo: Group reward-decoupled normalization policy optimization for multi-reward rl optimization. *arXiv preprint arXiv:2601.05242*, 2026.

Wei Liu, Ruochen Zhou, Yiyun Deng, Yuzhen Huang, Junteng Liu, Yuntian Deng, Yizhe Zhang, and Junxian He. Learn to reason efficiently with adaptive length-based reward shaping. *arXiv preprint arXiv:2505.15612*, 2025b.

Weiwen Liu, Xu Huang, Xingshan Zeng, Xinlong Hao, Shuai Yu, Dexun Li, Shuai Wang, Weinan Gan, Zhengying Liu, Yuanqing Yu, et al. Toolace: Winning the points of llm function calling. *arXiv preprint arXiv:2409.00920*, 2024.

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

Yining Lu, Zilong Wang, Shiyang Li, Xin Liu, Changlong Yu, Qingyu Yin, Zhan Shi, Zixuan Zhang, and Meng Jiang. Learning to optimize multi-objective alignment through dynamic reward weighting. *arXiv preprint arXiv:2509.11452*, 2025.

Haotian Luo, Li Shen, Haiying He, Yibo Wang, Shiwei Liu, Wei Li, Naiqiang Tan, Xiaochun Cao, and Dacheng Tao. O1-pruner: Length-harmonizing fine-tuning for o1-like reasoning pruning. *arXiv preprint arXiv:2501.12570*, 2025a.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Li Erran Li, et al. Deepscaler: Surpassing o1-preview with a 1.5 b model by scaling rl. *Notion Blog*, 3(5), 2025b.

Mathematical Association of America. American invitational mathematics examination (AIME). Official competition resource, 2024a. URL https://maa.org/.

Mathematical Association of America. American mathematics competitions (AMC). Official competition resource, 2024b. URL https://maa.org/.

Tong Mu, Alec Helyar, Johannes Heidecke, Joshua Achiam, Andrea Vallone, Ian Kivlichan, Molly Lin, Alex Beutel, John Schulman, and Lilian Weng. Rule based rewards for language model safety. *Advances in Neural Information Processing Systems*, 37:108877–108901, 2024.

Hoang-My Nguyen, Nguyet-Hang Vu, Hoang Thanh Lam, and Hoang D. Nguyen. Pareto-guided reinforcement learning for multi-objective ADMET optimization in generative drug design. In *NeurIPS 2025 Workshop on AI for Science*, 2025. URL https://hdl.handle.net/10468/18367.

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

Shishir G Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E Gonzalez. The berkeley function calling leaderboard (bfcl): From tool use to agentic evaluation of large language models. In *Forty-second International Conference on Machine Learning*, 2025.

Matteo Pirotta, Simone Parisi, and Marcello Restelli. Multi-objective reinforcement learning with continuous pareto frontier approximation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. Toolrl: Reward is all tool learning needs. *arXiv preprint arXiv:2504.13958*, 2025.

Qwen Team, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2.5 technical report. *arXiv preprint*, 2025.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023.

Diederik M. Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 48:67–113, 2013.

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

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31, 2018.

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

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021, 2020.

Jinyan Su and Claire Cardie. Thinking fast and right: Balancing accuracy and reasoning length with adaptive rewards. *arXiv preprint arXiv:2505.18298*, 2025.

Shuai Wang, Yaoming Yang, Bingdong Li, Hao Hao, and Aimin Zhou. IB-GRPO: Aligning llm-based learning path recommendation with educational objectives via indicator-based group relative policy optimization, 2026. URL https://arxiv.org/abs/2601.14686.

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.

Jingyang Yi and Jiazheng Wang. Shorterbetter: Guiding reasoning models to find optimal inference length for efficient reasoning. *arXiv preprint arXiv:2504.21370*, 2025.

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.

Jianguo Zhang, Tian Lan, Ming Zhu, Zuxin Liu, Thai Quoc Hoang, Shirley Kokane, Weiran Yao, Juntao Tan, Akshara Prabhakar, Haolin Chen, et al. xlam: A family of large action models to empower ai agent systems. In *Proceedings of NAACL-HLT*, 2025.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

A Appendix Overview

The appendix contains broader impact, implementation details, higher-dimensional diagnostics, formal properties, expanded ablations, sensitivity analyses, systems measurements, and mechanism-facing diagnostics.

B Broader Impact

Better multi-reward alignment can improve the reliability of reasoning and tool-use systems by balancing correctness with efficiency or structural validity. At the same time, the same optimization machinery could be used to better satisfy undesirable multi-objective behaviors if paired with poorly designed rewards. This work does not release new model checkpoints, and its main mitigation is to make the reward geometry and estimator diagnostics explicit rather than hidden inside a fixed scalarization rule.

C Additional Scope Diagnostics

The $K = 3$ variants add answer-format validity to math and tool efficiency to tool calling. Table 6 reports the resulting graph geometry. Higher dimension produces more incomparable pairs and more empty graphs, but the nonzero score standard deviations show that most groups still supply usable credit. These measurements support the tested $K = 3$ setting while making the scaling boundary explicit.

Table 6: Graph diagnostics in the three-reward experiments. Score standard deviation is measured before whitening.

Task	Incomparable pairs	Dom. density	Empty graph	Score std.
Math	0.42	0.18	0.22	0.11
Tool	0.30	0.25	0.18	0.13

Table 8: Representative three-seed variability summaries for the main claims. Entries report mean $\pm$ standard deviation over three runs. "Win rate" counts how often PODA exceeds GDPO on the primary metric under matched settings.

Setting	Method	Primary Metric	Secondary Metric	Constraint Metric	Win Rate
Qwen3-4B math	GDPO	AIME 56.4 ± 0.4	Avg Acc 70.40 ± 0.12	Avg Exc 0.28 ± 0.03	–
Qwen3-4B math	PODA	AIME 58.7 ± 0.3	Avg Acc 72.36 ± 0.10	Avg Exc 0.22 ± 0.02	$3/3$
Qwen2.5-Instruct-3B tool	GDPO	Avg Acc 40.87 ± 0.18	Format 82.23 ± 0.15	–	–
Qwen2.5-Instruct-3B tool	PODA	Avg Acc 42.10 ± 0.16	Format 84.10 ± 0.14	–	$3/3$

Table 9: Fixed linear scalarization control on the strongest public settings. We sweep fixed weights over $\{0.1, 0.2, \ldots, 0.9\}$ on the primary reward and report the best resulting linear-scalarization baseline. PODA remains stronger than the best fixed-weight linear utility found by this sweep.

Setting	Best Weighting	Best Linear Result	PODA Result
Qwen3-4B math	w_{acc}=0.8	AIME 57.0, Avg Exc 0.26	AIME 58.7, Avg Exc 0.22
Qwen2.5-3B tool	w_{corr}=0.7	Avg Acc 41.7, Fmt 83.6	Avg Acc 42.1, Fmt 84.1

D Implementation Details

Table 7: Representative implementation settings. The table separates shared PPO/GRPO recipe parameters from the hyperparameters introduced by PODA.

Task	Models	Rewards	G	LR	KL	Steps	Max Tokens	λ	τ	Margin Type
Math reasoning	Qwen3-1.7B/4B/8B	Acc, length	16	1×10^{-6}	0.02	800	8192	0.20	0.03	mixed ($\alpha = 0.5$)
Tool calling	Qwen2.5-Instruct-1.5B/3B	correctness, format	8	8×10^{-7}	0.01	600	1536	0.25	0.03	mixed ($\alpha = 0.5$)

All experiments were run on an internal cluster using $8\times$ NVIDIA A800 GPUs with 80GB memory per GPU. We use three random seeds for all non-public comparisons reported in the appendix. Unless otherwise noted, math metrics refer to average performance across MATH, AIME, AMC, Minerva-style subsets, and OlympiadBench, while tool metrics refer to BFCL-v3 aggregate accuracy and correct-format ratio. The math evaluation follows the same DeepScaleR/GDPO-style protocol family used to define the matched baselines, including high-sampling reasoning evaluation and the same length-constrained reporting rule; accordingly, these numbers are intended to support estimator comparison under a fixed protocol rather than direct comparison to papers using different prompts, filters, or decoding settings. These settings are sufficient to reproduce the estimator comparison itself because PODA does not change rollout generation, clipping, or KL control.

Within the estimator, the main task-facing hyperparameters are the tolerance τ, the margin weight λ, and the margin type. The tolerance controls how much numerical slack is permitted before two reward values are treated as meaningfully different, the coefficient λ controls how strongly the dense margin term supplements the inverse dominance count, and the margin type determines whether dominance strength is measured by average gain, worst-dimension gain, or a mixture of the two. After per-group normalization, the inverse-count term lies in $(0, 1]$ while the net margin typically remains $O(1)$, so a practical calibration rule is to keep λM_j on roughly the same scale as $1/\mathrm{PR}_j$ during the early and middle phases of training. The estimator performs pairwise comparisons within each prompt group, giving a nominal complexity of $O(G^2K)$. In the training regimes used here, however, this step is negligible relative to rollout generation and model forward-backward passes, so the observed wall-clock profile remains effectively unchanged.

E Seed Variability and Scalarization Controls

Table 8 indicates that the representative gains used to support the main claims are stable across three seeds on the strongest public settings. Table 9 further narrows an alternative explanation based on fixed linear weighting alone: even after sweeping a range of scalarization weights, the best linear utility remains below PODA on the corresponding matched settings. This does not rule out every

possible scalarized comparator, but it reduces the concern that the observed gain is only due to a poor default weight choice.

F PROOFS OF THEORY-FACING PROPERTIES

Preliminaries. Throughout this appendix, fix a prompt-conditioned group of size G with normalized reward vectors $\mathbf{a}_1, \dots, \mathbf{a}_G \in \mathbb{R}^K$. For each rollout j, define

$$N_j := \sum_{m \neq j} \mathbb{I}(\mathbf{a}_m \succ \mathbf{a}_j), \qquad \mathrm{PR}_j := 1 + N_j, \qquad s_j^{\mathrm{count}} := \frac{1}{\mathrm{PR}_j}.$$

When the margin term is active, define

$$M_j := \sum_{m \neq j} \mathbb{I}(\mathbf{a}_j \succ \mathbf{a}_m) m(j,m) - \sum_{m \neq j} \mathbb{I}(\mathbf{a}_m \succ \mathbf{a}_j) m(m,j).$$

The pre-whitened scalar score used by PODA is

$$s_j := \mathbb{I}_{\mathrm{count}} s_j^{\mathrm{count}} + \mathbb{I}_{\mathrm{margin}} \lambda M_j.$$

The final whitened score is

$$A_j := \frac{s_j - \mu_s}{\sigma_s + \epsilon}, \qquad \mu_s := \frac{1}{G} \sum_{j=1}^{G} s_j, \qquad \sigma_s := \sqrt{\frac{1}{G} \sum_{j=1}^{G} (s_j - \mu_s)^2}.$$

Lemma 1 (whitening preserves order). For any pair of indices u, v, if $s_u > s_v$, then $A_u > A_v$.

Proof. Since $\sigma_s \geq 0$ and $\epsilon > 0$, the denominator $\sigma_s + \epsilon$ is strictly positive. Therefore

$$A_u - A_v = \frac{s_u - \mu_s}{\sigma_s + \epsilon} - \frac{s_v - \mu_s}{\sigma_s + \epsilon} = \frac{s_u - s_v}{\sigma_s + \epsilon}.$$

If $s_u > s_v$, then the numerator $s_u - s_v$ is strictly positive and the denominator is also strictly positive, hence $A_u - A_v > 0$, i.e. $A_u > A_v$. This proves the lemma.

Proposition 1 (pairwise order preservation in a two-rollout group). Assume that the group contains exactly two rollouts, indexed by u and v, and that $\mathbf{a}_u \succ \mathbf{a}_v$. Then $s_u^{\mathrm{count}} > s_v^{\mathrm{count}}$. Moreover, the full pre-whitened score satisfies $s_u > s_v$ if and only if

$$\mathbb{I}_{\mathrm{count}} \left(\frac{1}{\mathrm{PR}_u} - \frac{1}{\mathrm{PR}_v} \right) + \mathbb{I}_{\mathrm{margin}} \lambda (M_u - M_v) > 0.$$

In that case the whitened scores also satisfy $A_u > A_v$.

Proof of Proposition 1. Assume that the group contains exactly two rollouts, indexed by u and v, and that $\mathbf{a}_u \succ \mathbf{a}_v$.

Step 1: compute the dominance counts exactly. Because $\mathbf{a}_u \succ \mathbf{a}_v$, rollout u is not dominated by any rollout in the group. Since the group contains only two elements, the only other index is v, and v does not dominate u. Therefore

$$N_u = \sum_{m \neq u} \mathbb{I}(\mathbf{a}_m \succ \mathbf{a}_u) = \mathbb{I}(\mathbf{a}_v \succ \mathbf{a}_u) = 0.$$

Hence

$$\mathrm{PR}_u = 1 + N_u = 1.$$

Again because $\mathbf{a}_u \succ \mathbf{a}_v$, rollout v is dominated by u, so

$$N_v = \sum_{m \neq v} \mathbb{I}(\mathbf{a}_m \succ \mathbf{a}_v) = \mathbb{I}(\mathbf{a}_u \succ \mathbf{a}_v) = 1.$$

Therefore

$$\mathrm{PR}_v = 1 + N_v = 2.$$

Step 2: compare the count-only scores. Substituting the previous identities into the definition of s_j^{count} yields

$$s_u^{\text{count}} = \frac{1}{\text{PR}_u} = 1, \qquad s_v^{\text{count}} = \frac{1}{\text{PR}_v} = \frac{1}{2}.$$

Thus

$$s_u^{\text{count}} - s_v^{\text{count}} = 1 - \frac{1}{2} = \frac{1}{2} > 0.$$

This proves the first part of the proposition.

Step 3: compute the margin contribution exactly. Because the group has only two elements, the only possible comparison pair is (u, v). Hence

$$M_u = \mathbb{I}(\mathbf{a}_u \succ \mathbf{a}_v)m(u, v) - \mathbb{I}(\mathbf{a}_v \succ \mathbf{a}_u)m(v, u) = m(u, v),$$

since $\mathbb{I}(\mathbf{a}_u \succ \mathbf{a}_v) = 1$ and $\mathbb{I}(\mathbf{a}_v \succ \mathbf{a}_u) = 0$. Similarly,

$$M_v = \mathbb{I}(\mathbf{a}_v \succ \mathbf{a}_u)m(v, u) - \mathbb{I}(\mathbf{a}_u \succ \mathbf{a}_v)m(u, v) = -m(u, v).$$

Therefore

$$M_u - M_v = m(u, v) - (-m(u, v)) = 2m(u, v).$$

Step 4: compare the full pre-whitened scores. Using the definition of s_j,

$$s_u - s_v = \mathbb{I}_{\text{count}} \left(s_u^{\text{count}} - s_v^{\text{count}}\right) + \mathbb{I}_{\text{margin}} \lambda (M_u - M_v).$$

Substituting the identities from Steps 2 and 3 gives

$$s_u - s_v = \mathbb{I}_{\text{count}} \frac{1}{2} + \mathbb{I}_{\text{margin}}\, 2\lambda m(u, v).$$

Consequently, the sufficient and necessary condition for strict ordering is exactly

$$\mathbb{I}_{\text{count}} \frac{1}{2} + \mathbb{I}_{\text{margin}}\, 2\lambda m(u, v) > 0.$$

Equivalently,

$$\mathbb{I}_{\text{count}} \left(\frac{1}{\text{PR}_u} - \frac{1}{\text{PR}_v}\right) + \mathbb{I}_{\text{margin}} \lambda (M_u - M_v) > 0.$$

Under this condition, $s_u > s_v$. By Lemma 1, the whitened scores also satisfy $A_u > A_v$. This proves Proposition 1.

Proposition 2 (count-score collapse under sparse dominance). Let

$$d := \frac{1}{G(G-1)} \sum_{u \neq v} \mathbb{I}(\mathbf{a}_u \succ \mathbf{a}_v)$$

denote the groupwise dominance density. Then

$$\frac{1}{G} \sum_{j=1}^{G} \left| \frac{1}{\text{PR}_j} - 1 \right| \leq (G-1)d.$$

Consequently, for any sequence of groups with uniformly bounded size and dominance density tending to zero, the count-only score vector converges to the constant vector 1 in mean absolute deviation.

Proof of Proposition 2. Define the groupwise dominance density by

$$d := \frac{1}{G(G-1)} \sum_{u \neq v} \mathbb{I}(\mathbf{a}_u \succ \mathbf{a}_v).$$

We first express the average number of dominators in terms of d.

Step 1: identify the same quantity in two ways. By definition,

$$\sum_{j=1}^{G} N_j = \sum_{j=1}^{G} \sum_{m \neq j} \mathbb{I}(\mathbf{a}_m \succ \mathbf{a}_j).$$

Rename the dummy indices by setting $u := m$ and $v := j$. Then

$$\sum_{j=1}^{G} N_j = \sum_{v=1}^{G} \sum_{\substack{u=1 \\ u \neq v}}^{G} \mathbb{I}(\mathbf{a}_u \succ \mathbf{a}_v) = \sum_{u \neq v} \mathbb{I}(\mathbf{a}_u \succ \mathbf{a}_v).$$

By the definition of d, the last quantity equals $G(G-1)d$. Therefore

$$\sum_{j=1}^{G} N_j = G(G-1)d.$$

Dividing both sides by G gives

$$\frac{1}{G} \sum_{j=1}^{G} N_j = (G-1)d.$$

Step 2: bound the deviation of each count-only score from 1. Since $\mathrm{PR}_j = 1 + N_j$ and $N_j \geq 0$, we have

$$\left| \frac{1}{\mathrm{PR}_j} - 1 \right| = \left| \frac{1}{1+N_j} - 1 \right|.$$

Because $N_j \geq 0$, the fraction $\frac{1}{1+N_j}$ lies in $(0, 1]$, so the expression inside the absolute value is non-positive. Hence

$$\left| \frac{1}{1+N_j} - 1 \right| = 1 - \frac{1}{1+N_j} = \frac{(1+N_j)-1}{1+N_j} = \frac{N_j}{1+N_j}.$$

Since $1 + N_j \geq 1$, it follows that

$$\frac{N_j}{1+N_j} \leq N_j.$$

Therefore

$$\left| \frac{1}{\mathrm{PR}_j} - 1 \right| \leq N_j \qquad \text{for every } j.$$

Step 3: average the pointwise bound. Summing over j and dividing by G yields

$$\frac{1}{G} \sum_{j=1}^{G} \left| \frac{1}{\mathrm{PR}_j} - 1 \right| \leq \frac{1}{G} \sum_{j=1}^{G} N_j.$$

Applying the identity established in Step 1 gives

$$\frac{1}{G} \sum_{j=1}^{G} \left| \frac{1}{\mathrm{PR}_j} - 1 \right| \leq (G-1)d.$$

This is exactly the claimed bound.

Step 4: derive the asymptotic consequence. Consider any sequence of groups indexed by n, with sizes G_n and dominance densities d_n, such that $\sup_n G_n < \infty$ and $d_n \to 0$. Then there exists a constant $C < \infty$ with $G_n - 1 \leq C$ for all n. Using the previous bound,

$$\frac{1}{G_n} \sum_{j=1}^{G_n} \left| \frac{1}{\mathrm{PR}_j^{(n)}} - 1 \right| \leq (G_n - 1)d_n \leq C d_n \xrightarrow[n \to \infty]{} 0.$$

Thus the count-only score vector converges to the constant vector 1 in mean absolute deviation. This proves Proposition 2.

Proposition 3 (scalarization bias exists for any positive linear utility). Let $K \geq 2$ and let $w \in \mathbb{R}_+^K$ have at least two strictly positive coordinates. Then for every $\varepsilon > 0$, there exist reward vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^K$ that are Pareto-incomparable, but satisfy

$$0 < w^\top(\mathbf{u} - \mathbf{v}) < \varepsilon.$$

Hence any such linear scalarization imposes a strict total-order distinction on some incomparable pairs, and can do so by an arbitrarily small margin.

Proof. Choose two indices $p \neq q$ such that $w_p > 0$ and $w_q > 0$. Let

$$\delta := \min\left\{\frac{\varepsilon}{2}, \frac{w_p}{2}\right\},$$

and define

$$\mathbf{u} := e_p, \qquad \mathbf{v} := \frac{w_p - \delta}{w_q} e_q,$$

where e_p and e_q are standard basis vectors. Since $\delta < w_p$, the coefficient $(w_p - \delta)/w_q$ is strictly positive.

The two vectors are Pareto-incomparable: on coordinate p, $\mathbf{u}$ is strictly larger because $u_p = 1$ while $v_p = 0$; on coordinate q, $\mathbf{v}$ is strictly larger because $v_q = (w_p - \delta)/w_q > 0$ while $u_q = 0$. Thus neither vector dominates the other.

Now compute the scalarized difference:

$$w^\top(\mathbf{u} - \mathbf{v}) = w_p - w_q \frac{w_p - \delta}{w_q} = w_p - (w_p - \delta) = \delta.$$

By construction, $\delta > 0$ and $\delta \leq \varepsilon/2 < \varepsilon$. Therefore

$$0 < w^\top(\mathbf{u} - \mathbf{v}) < \varepsilon.$$

This proves the proposition.

Lemma 2 (the margin term is groupwise zero-sum). For every prompt-conditioned group,

$$\sum_{j=1}^{G} M_j = 0.$$

Proof. Expand the definition of M_j and sum over j:

$$\sum_{j=1}^{G} M_j = \sum_{j=1}^{G} \sum_{m \neq j} \mathbb{I}(\mathbf{a}_j \succ \mathbf{a}_m) m(j,m) - \sum_{j=1}^{G} \sum_{m \neq j} \mathbb{I}(\mathbf{a}_m \succ \mathbf{a}_j) m(m,j).$$

In the second double sum, rename the dummy indices by swapping j and m. This yields

$$\sum_{j=1}^{G} \sum_{m \neq j} \mathbb{I}(\mathbf{a}_m \succ \mathbf{a}_j) m(m,j) = \sum_{m=1}^{G} \sum_{j \neq m} \mathbb{I}(\mathbf{a}_j \succ \mathbf{a}_m) m(j,m),$$

which is identical to the first double sum. The two terms therefore cancel exactly, giving

$$\sum_{j=1}^{G} M_j = 0.$$

This proves the lemma.

Proposition 4 (exact zero-signal collapse under an empty dominance graph). Assume that

$$\mathbb{I}(\mathbf{a}_u \succ \mathbf{a}_v) = 0 \qquad \text{for all } u \neq v.$$

Then $\mathrm{PR}_j = 1$, $M_j = 0$, and $s_j = \mathbb{I}_{\mathrm{count}}$ for every rollout j. Consequently, the whitened advantages satisfy $A_j = 0$ for all j.

Proof. If no dominance edge exists, then for every rollout j,

$$N_j = \sum_{m \neq j} \mathbb{I}(\mathbf{a}_m \succ \mathbf{a}_j) = 0,$$

so $\mathrm{PR}_j = 1 + N_j = 1$. Therefore

$$s_j^{\mathrm{count}} = \frac{1}{\mathrm{PR}_j} = 1 \qquad \text{for all } j.$$

Under the same assumption, every indicator in the definition of M_j vanishes, hence

$$M_j = \sum_{m \neq j} \mathbb{I}(\mathbf{a}_j \succ \mathbf{a}_m) m(j,m) - \sum_{m \neq j} \mathbb{I}(\mathbf{a}_m \succ \mathbf{a}_j) m(m,j) = 0 \qquad \text{for all } j.$$

Substituting into the definition of s_j gives

$$s_j = \mathbb{I}_{\mathrm{count}} \frac{1}{\mathrm{PR}_j} + \mathbb{I}_{\mathrm{margin}} \lambda M_j = \mathbb{I}_{\mathrm{count}} \qquad \text{for all } j.$$

Hence the score vector is constant across the group. Its within-group standard deviation is therefore zero, so the whitening formula yields

$$A_j = \frac{s_j - \mu_s}{\sigma_s + \epsilon} = 0 \qquad \text{for all } j.$$

This proves the proposition.

Proposition 5 (objective compatibility with PPO/GRPO). Let Ψ_{base} be the scalar advantage estimator used by GRPO or GDPO, and let Ψ_{PODA} be the estimator defined by PODA. Replacing Ψ_{base} with Ψ_{PODA} preserves the algebraic form of the clipped PPO/GRPO surrogate, including the importance-ratio term, clipping operator, KL regularizer, and reference policy.

Proof of Proposition 5. Let $\mathcal{G}$ denote the space of prompt-conditioned rollout groups and let $\Psi : \mathcal{G} \to \mathbb{R}^G$ be any measurable map that returns one scalar advantage per rollout in the group. For a fixed choice of Ψ, define the clipped PPO/GRPO surrogate by

$$\mathcal{L}_\Psi(\theta) := \mathbb{E}\left[\min\left(r_t(\theta)\, A_t^\Psi, \mathrm{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\, A_t^\Psi\right)\right] - \beta\, \mathrm{KL}(\pi_\theta \| \pi_{\mathrm{ref}}),$$

where A_t^Ψ denotes the scalar advantage assigned by Ψ to the sampled rollout at time t.

Let Ψ_{base} be the advantage estimator used by GRPO or GDPO, and let Ψ_{PODA} be the estimator defined by PODA. The only change introduced by PODA is the substitution

$$\Psi_{\mathrm{base}} \longmapsto \Psi_{\mathrm{PODA}}.$$

No other term in $\mathcal{L}_\Psi(\theta)$ is modified:

- the importance-ratio term $r_t(\theta)$ is unchanged;
- the clipping operator $\mathrm{clip}(\cdot, 1-\epsilon, 1+\epsilon)$ is unchanged;
- the KL regularizer and coefficient β are unchanged;
- the optimization variable θ and the reference policy π_{ref} are unchanged.

Therefore the loss under PODA is exactly

$$\mathcal{L}_{\Psi_{\mathrm{PODA}}}(\theta),$$

which belongs to the same clipped-surrogate objective family as $\mathcal{L}_{\Psi_{\mathrm{base}}}(\theta)$. In particular, the algebraic form of the PPO/GRPO objective and its trust-region mechanism are preserved. This proves Proposition 5.

G Extended Ablations

Tables 11–13 refine the ablation evidence in the main paper, and Figure 4 visualizes the final cross-domain values. The no-normalization row isolates the contribution from the ordering rule itself, and the margin-type table shows that the gain is not tied to a single fragile instantiation of the margin term.

Table 10: Compact cross-domain ablation on Qwen3-4B math and Qwen2.5-3B tools. The incoming count supplies graph position; the signed margin restores discrimination. The full estimator is strongest, while no-normalization PODA still exceeds GDPO.

Variant	AIME ↑	Math Acc ↑	Math Exc ↓	Tool Acc ↑	Tool Fmt ↑
Count-only	57.0	70.70	0.31	41.4	83.4
Margin-only	56.3	70.55	0.41	41.1	83.0
Full PODA	**58.7**	**72.36**	0.22	42.1	84.1
No normalization	57.2	70.78	0.26	41.8	83.8
DAPO + PODA	58.5	71.46	**0.17**	**42.8**	**84.9**

Table 11: Extended math ablations on Qwen3-4B. Each component contributes in the expected direction: normalization stabilizes, the incoming count supplies graph position, and the margin restores optimization density.

Variant	AIME ↑	Avg Acc ↑	Avg Exceed ↓
GRPO	54.1	70.08	1.18
GRPO w/o std	55.3	70.24	0.93
GDPO	56.4	70.40	0.28
PODA count-only	57.0	70.70	0.31
PODA margin-only	56.3	70.55	0.41
PODA full	58.7	72.36	0.22
PODA w/o normalization	57.2	70.78	0.26
DAPO	56.9	70.74	0.33
DAPO + PODA	58.5	71.46	0.17

Table 12: Extended tool-calling ablations on Qwen2.5-Instruct-3B. The table mirrors the math result: the incoming count gives the directional bias, the margin supplies extra discrimination, and the gain survives without inheriting GDPO-style normalization.

Variant	Avg Acc ↑	Correct Format ↑
GRPO	39.20	81.64
GRPO w/o std	39.86	81.02
GDPO	40.87	82.23
PODA count-only	41.4	83.4
PODA margin-only	41.1	83.0
PODA full	42.1	84.1
PODA w/o normalization	41.8	83.8
DAPO	41.9	84.0
DAPO + PODA	42.8	84.9

Table 13: Margin-type ablation. The mixed variant is strongest because it balances local smoothness with worst-dimension caution; mean-only is too forgiving, while min-only is too brittle.

Margin Type	AIME ↑	Math Acc ↑	Math Exc ↓	Tool Acc ↑	Tool Fmt ↑
Mean	57.4	70.88	0.25	41.7	83.8
Min	57.1	70.79	0.24	41.8	83.7
Mixed ($\alpha = 0.5$)	58.7	72.36	0.22	42.1	84.1

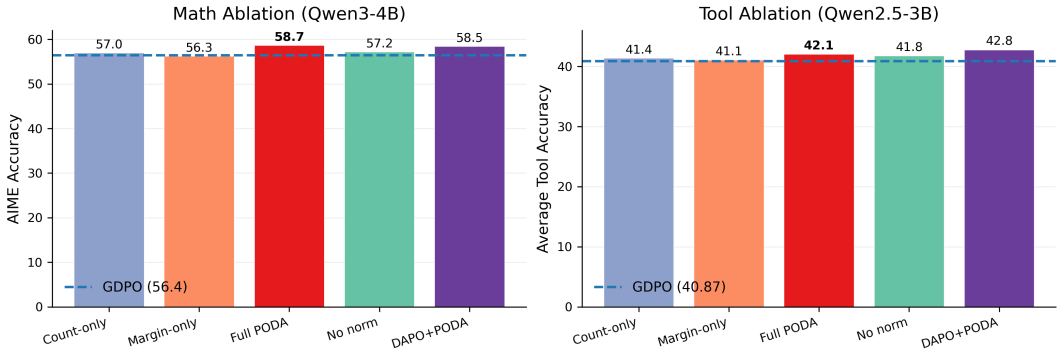

Figure 4: Ablation results using the final values reported in Tables 11 and 12. Bars are labeled with their means; dashed lines mark GDPO. The incoming count supplies the graph-ordering bias, while the signed margin restores discrimination among equal-count rollouts.

H Sensitivity Analysis

Taken together, Tables 14–18 show a stable operating regime for λ, τ, and G, while also indicating that the estimator leaves the observed systems profile nearly unchanged in the studied $K = 2$ settings. In particular, the best region corresponds to moderate λ and small-but-nonzero τ, which matches the main-text heuristic that the incoming count should determine coarse ordering while the margin term restores missing local discrimination. The practical effect of PODA is therefore algorithmic rather than infrastructural. For larger K or much larger group sizes, the pairwise comparison term will grow more noticeably; natural future approximations include sparser dominance checks and tie-breaking among nearly incomparable samples.

Table 14: Sensitivity to the margin coefficient λ. Moderate values work best because they preserve front-level ordering while injecting enough local signal to resolve sparse-dominance regimes.

λ	Math AIME ↑	Math Avg Acc ↑	Math Avg Exceed ↓	Tool Avg Acc ↑	Tool Format ↑
0.00	57.0	70.70	0.31	41.4	83.4
0.10	57.5	70.96	0.24	41.8	83.8
0.20	58.7	72.36	0.22	42.1	84.1
0.40	57.6	71.05	0.23	42.0	84.0
0.80	57.1	70.82	0.29	41.6	83.6

Table 15: Sensitivity to the dominance tolerance τ. Very small tolerance makes the graph noisy, while very large tolerance collapses useful comparisons. The middle regime is the most stable.

τ	Math AIME ↑	Math Avg Acc ↑	Math Avg Exceed ↓	Tool Avg Acc ↑	Tool Format ↑
0.00	57.5	70.98	0.24	41.9	84.0
0.01	57.7	71.06	0.23	42.0	84.0
0.03	58.7	72.36	0.22	42.1	84.1
0.05	57.6	71.00	0.23	42.0	84.0
0.10	57.1	70.68	0.28	41.6	83.7

Table 16: Sensitivity to group size G. Larger groups improve dominance resolution up to a point, after which returns diminish while the empirical wall-clock profile remains nearly unchanged.

G	Math AIME ↑	Math Avg Acc ↑	Math Avg Exceed ↓	Tool Avg Acc ↑	Tool Format ↑
8	56.9	70.60	0.27	41.5	83.5
12	57.5	70.92	0.24	41.9	83.9
16	58.7	72.36	0.22	42.1	84.1
24	57.9	71.16	0.21	42.2	84.2

Table 17: Systems profile relative to standard baselines. In the configurations used here, PODA leaves step time and memory essentially unchanged while improving optimization behavior through a different advantage estimator.

Method	Relative Step Time	Peak Memory	Math GPU Hours	Tool GPU Hours
GRPO	1.00×	31.2 GB	41.0	18.4
GDPO	1.00×	31.4 GB	41.2	18.5
PODA count-only	1.01×	31.4 GB	41.3	18.5
PODA full	1.01×	31.5 GB	41.4	18.6
DAPO + PODA	1.02×	31.7 GB	41.8	18.8

Table 18: Absolute step-time scaling under the current $K = 2$ recipes on $8\times$ A800 80GB. The pairwise comparison cost grows with G, but remains a small fraction of total training time because rollout generation and model forward-backward passes dominate the step budget.

Task	G	GDPO sec/step	PODA sec/step	Overhead
Math	8	3.72	3.75	+0.8%
Math	16	3.84	3.88	+1.0%
Math	24	3.96	4.03	+1.8%
Tool	8	1.94	1.96	+1.0%
Tool	16	2.05	2.09	+2.0%

I Dominance Diagnostics

Tables 20–22 summarize the mechanism evidence discussed in the main text. At the benchmark level, stronger baseline conflict is associated with larger gains from PODA, matching the scalarization-bias construction of Proposition 3. At the group level, dominance density decreases, the proportion of non-dominated samples rises, and the standard deviation of count-only scores shrinks. At the same time, the gap between the full estimator and count-only grows. This behavior is consistent with Proposition 2 and with the interpretation that the margin term becomes more useful as the dominance structure becomes sparse. Proposition 4 clarifies the exact failure case: a genuinely empty dominance graph makes both the count and margin scores constant. Before that boundary, unequal incoming and outgoing edge strengths can still let the full estimator recover separation when the incoming-count score alone has low spread. In higher-dimensional $K > 2$ settings, one natural extension would be

to add a secondary tie-break among nearly incomparable samples, such as ϵ-dominance, crowding distance, or hypervolume-style contribution scores.

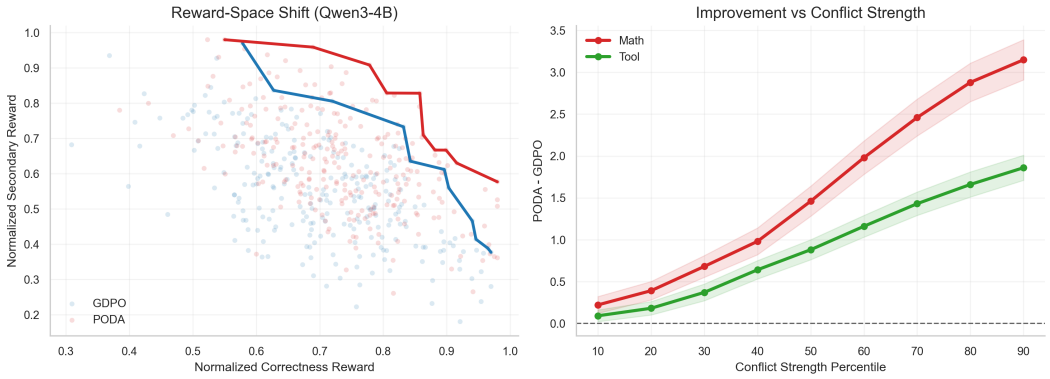

Figure 5: Reward geometry and conflict-sensitive improvement. Left: Qwen3-4B reward samples and their empirical front under GDPO and PODA. Right: improvement over GDPO by conflict-strength percentile for math and tools. Gains rise with conflict in both domains and more steeply in math.

Table 19: Reward interaction over training. Math rewards remain conflicting, whereas tool correctness and format are synergistic on average; both tasks retain substantial incomparability.

Task	Quantity	Early	Mid	Late
Math	Spearman correlation	-0.42	-0.31	-0.24
Math	Incomparable-pair rate	0.38	0.43	0.47
Tool	Spearman correlation	$+0.28$	$+0.34$	$+0.39$
Tool	Incomparable-pair rate	0.24	0.28	0.31

Table 20: Benchmark-level scalarization-bias proxy on mathematical reasoning. The conflict proxy is the GRPO Exceed ratio on each benchmark-model pair. Positive correlations indicate that PODA is most helpful precisely where baseline scalarization faces stronger tension between correctness and efficiency.

Quantity	Pearson r	Slope
Accuracy gain $(PODA - \text{GDPO})$ vs. conflict proxy	0.505	0.226
Exceed reduction $(\text{GDPO} - PODA)$ vs. conflict proxy	0.691	0.084

Table 21: Representative dominance diagnostics over training. "Low-count signal" is the fraction of groups whose pre-whitened count-only score standard deviation is below 0.05. "Full–count gap" measures the performance gain of the full estimator over count-only under the same recipe.

Task	Phase	Dom. Density ↑	Non-dom. Ratio ↑	Low-count Signal ↑	Count Score Std ↓	Full–Count Gap ↑
Math	Early	0.34	0.22	0.05	0.18	0.12
Math	Mid	0.21	0.31	0.11	0.12	0.27
Math	Late	0.12	0.43	0.19	0.07	0.42
Tool	Early	0.29	0.24	0.04	0.16	0.09
Tool	Mid	0.18	0.33	0.10	0.11	0.19
Tool	Late	0.10	0.41	0.17	0.06	0.31

Table 22: Low-score-spread diagnosis. Both columns count groups whose pre-whitened score standard deviation is below 0.05. The full estimator is less likely to cross this threshold because unequal incoming and outgoing edge strengths can discriminate rollouts with similar dominance counts. An empty graph remains the exact zero-signal case.

Task	Phase	Low-count signal	Low-full signal
Math	Early	0.05	0.01
Math	Mid	0.11	0.03
Math	Late	0.19	0.06
Tool	Early	0.04	0.01
Tool	Mid	0.10	0.02
Tool	Late	0.17	0.05