TGPO: TRACKED-GRADIENT POLICY OPTIMIZATION FOR EFFECTIVE MULTI-REWARD REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-reward reinforcement learning requires coordinating objectives whose signals differ in scale, variability, and direction. GDPO preserves reward distinctions through decoupled normalization, but reward statistics alone do not determine gradient compatibility. Sampling noise further affects nonlinear coordination across minibatches. We introduce Tracked-Gradient Policy Optimization (TGPO), which retains decoupled normalization and coordinates temporally tracked objective gradients. A projected recursive estimator maintains a bounded state for each objective. TGPO approximately maximizes the minimum predicted surrogate improvement within a relative neighborhood of the mean ascent direction. The same tracked vectors define both the interaction geometry and the gradient supplied to the optimizer, making temporal information part of the update to support stable gradient coordination. We evaluate TGPO against GDPO and DVAO on tool use, mathematical reasoning, and coding reasoning. Evaluation covers correctness metrics and constraint adherence. Compared with these baselines, TGPO achieves state-of-the-art aggregate performance across the evaluated settings, demonstrating the effectiveness and broad applicability of temporal gradient coordination in multi-reward reinforcement learning. Our code is available at https://anonymous.4open.science/r/TGPO-072E/.

1 INTRODUCTION

Reinforcement learning (RL) increasingly shapes language-model reasoning and behavior (Schulman et al., 2017; Shao et al., 2024; Guo et al., 2025). Post-training objectives extend beyond correctness to reasoning efficiency (Liu et al., 2026b; Aggarwal & Welleck, 2025), safety (Mu et al., 2024), and consistency between reasoning and answers (Chen et al., 2026). Practical applications combine these requirements, making multi-reward optimization central to post-training. Learning must extract informative objective signals and coordinate their effects on a shared policy.

Group Relative Policy Optimization (GRPO) (Shao et al., 2024) estimates relative advantages from responses to the same prompt without fitting a separate value model. DAPO and GSPO refine this framework for stability and efficiency (Yu et al., 2025; Zheng et al., 2025). Multiple rewards additionally require objective-aware normalization and aggregation. Summing raw rewards before group normalization can map distinct reward combinations to identical advantages, obscuring objective-specific information (Liu et al., 2026a).

Group reward-Decoupled Normalization Policy Optimization (GDPO) separately normalizes each reward within a response group, aggregates the resulting advantages, and applies batch normalization to control their scale (Liu et al., 2026a). Dynamic Variance-adaptive Advantage Optimization (DVAO) further adapts objective weights to empirical reward standard deviations within the sampled group (Jiang et al., 2026a). Both preserve reward distinctions during advantage construction, but combine the signals before policy differentiation. Their interaction in parameter space therefore remains implicit.

Advantages interact with policy scores to form gradients, which may align, oppose one another, or be orthogonal despite similar reward scales. First-order progress depends on these directions and the optimizer's executed displacement. As the policy evolves, minibatch noise can also produce

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.

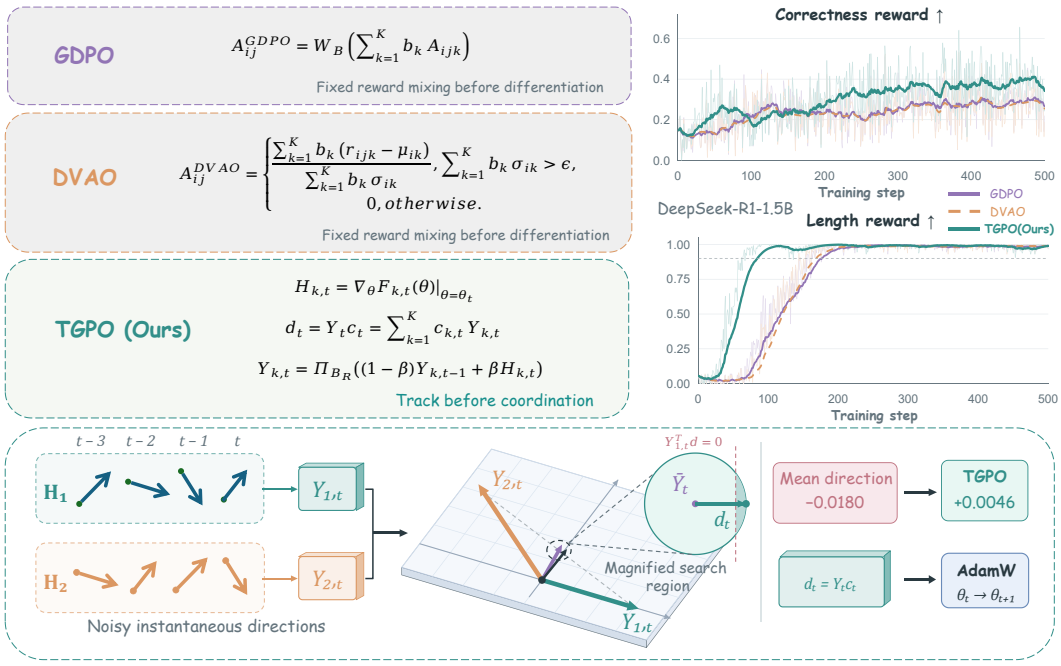

Figure 1: **Motivation: preserve objective directions before coordinating them.** Left: GDPO and DVAO mix reward advantages before differentiation; TGPO retains a temporal gradient state for each objective. Right: mathematical training illustrates the joint evolution of correctness and length rewards. Bottom: tracking carries directional history into local coordination, which turns a negative tracked margin positive in the illustrated construction. Appendix A.9 proves the displayed margin improvement.

abrupt changes in the coordination rule. Multi-reward learning therefore needs both an explicit representation of gradient geometry and a mechanism for retaining directional history.

We introduce **Tracked-Gradient Policy Optimization (TGPO)**, which extends reward decoupling to the evolving state of policy optimization. TGPO computes separate policy gradients from individually normalized advantages and maintains a bounded recursive state for each objective. It coordinates these states by maximizing the minimum predicted first-order improvement within a controlled neighborhood of their mean. The same tracked vectors define the interaction geometry and construct the gradient supplied to the optimizer. Objective-specific temporal information thus connects direction estimation to the shared policy update.

Our contributions are as follows:

- We characterize the gap between reward normalization and gradient compatibility, and derive bounds connecting temporal estimation error, mean-direction alignment, and actual optimizer displacement to local surrogate progress.
- We propose TGPO, integrating separately normalized policy surrogates, bounded recursive gradient states, and constrained max–min coordination. We derive a compact dual that computes the coordinated direction from objective interactions.
- Comparisons with GDPO and DVAO span three tasks and multiple model scales, jointly evaluating correctness and constraint adherence. Consistent gains in aggregate constraint adherence demonstrate the practical effectiveness of coordinating persistent objective directions.

2 FROM REWARD NORMALIZATION TO GRADIENT GEOMETRY

Context. Group-relative optimization (Shao et al., 2024) and practical training systems (Yu et al., 2025; Sheng et al., 2025) enable reinforcement learning with verifiable rewards. Multi-reward methods distinguish reward normalization from aggregation (Liu et al., 2026a; Jiang et al., 2026a); length-aware learning further studies how reward placement changes reasoning behavior (Jiang et al., 2026b). In parallel, gradient-based multi-objective learning explicitly optimizes local task interactions (Liu et al., 2021), and recursive estimation addresses noise in nonlinear gradient combination (Fernando et al., 2023). TGPO focuses on the policy-training interface between these two perspectives: preserving reward-specific direction information from advantage construction through persistent state to the gradient supplied to the optimizer.

Notation And Objective Signals. Let $\pi_\theta(y \mid x)$ be an autoregressive policy and $r_k(x, y)$ the kth of K rewards. The raw objectives are $J_k(\theta) = \mathbb{E}[r_k(x, y)]$. For prompt x_i, sample G responses, write $r_{ijk} = r_k(x_i, y_{ij})$, and compute

$$A_{ijk} = \frac{r_{ijk} - \mu_{ik}}{s_{ik} + \epsilon}, \qquad A_{ij}^{\mathrm{GDPO}} = \mathcal{W}_{\mathcal{B}}\left(\sum_k b_k A_{ijk}\right). \tag{1}$$

Here, μ_{ik} and s_{ik} are the group mean and sample standard deviation, b_k are fixed nonnegative base weights summing to one, and $\mathcal{W}_{\mathcal{B}}$ whitens the accepted batch. DVAO adapts its mixture to group standard deviations, $\widetilde{b}_{ik} = b_k\sigma_{ik}/\sum_\ell b_\ell\sigma_{i\ell}$. The local baseline uses population standard deviations σ_{ik} and the equivalent centered-reward form with a zero-variance gate. Appendix A.3 derives this equivalence and distinguishes scalar aggregation from separate clipping. Both methods construct a scalar advantage before differentiation. TGPO retains a separate advantage for each objective.

Where Compatibility Enters. For a smooth objective with gradient g_k, a small displacement ηd gives $J_k(\theta + \eta d) - J_k(\theta) = \eta g_k^\top d + O(\eta^2)$. For the linear combination $d = \sum_\ell a_\ell g_\ell$,

$$g_k^\top d = a_k \left\|g_k\right\|^2 + \sum_{\ell \neq k} a_\ell g_k^\top g_\ell. \tag{2}$$

The cross terms can outweigh the self term. For example, $g_1 = (1, 0)$ and $g_2 = (-2, 1)$ produce a mean direction $(-1/2, 1/2)$ that decreases objective 1 to first order. Reward magnitudes alone do not expose this geometry. For an empirical, unclipped normalized surrogate, collect the appropriately weighted token scores in S and its advantages in A_k. Its gradient $h_k = S^\top A_k$ satisfies $h_k^\top h_\ell = A_k^\top S S^\top A_\ell$. Advantage covariance omits the score kernel $SS^\top$. This distinction motivates measuring directions in parameter space.

Why Time Matters. Let $\mathcal{C}$ map a collection of gradient estimates to a coordinated direction. Because conflict resolution is nonlinear, $\mathbb{E}[\mathcal{C}(\widehat{g}_1, \ldots, \widehat{g}_K)]$ generally differs from $\mathcal{C}(\mathbb{E}\widehat{g}_1, \ldots, \mathbb{E}\widehat{g}_K)$. Increasing the resolution of scalar advantages does not remove this effect. TGPO moves temporal estimation before geometric coordination, while retaining the resulting vector state in the update. Figure 1 summarizes the conceptual progression.

3 TRACKED-GRADIENT POLICY OPTIMIZATION

3.1 PRESERVING OBJECTIVE IDENTITY IN THE POLICY SURROGATE

Broadcast A_{ijk} over response tokens and subtract its valid-token mean over the complete accepted rollout batch. Denote the fixed result by $\widehat{A}_{k,z}$, where z indexes valid response tokens in actor minibatch t. With importance ratio $q_z(\theta) = \pi_\theta(y_z \mid x, y_{<z})/\pi_{\mathrm{old}}(y_z \mid x, y_{<z})$, define

$$\begin{aligned} F_{k,t}(\theta) &= \frac{1}{N_t}\sum_{z=1}^{N_t} \min\Big\{q_z(\theta)\widehat{A}_{k,z},\ \mathrm{clip}(q_z(\theta), 1-\epsilon_-, 1+\epsilon_+)\widehat{A}_{k,z}\Big\}, \\ H_{k,t} &= \nabla_\theta F_{k,t}(\theta_t). \end{aligned} \tag{3}$$

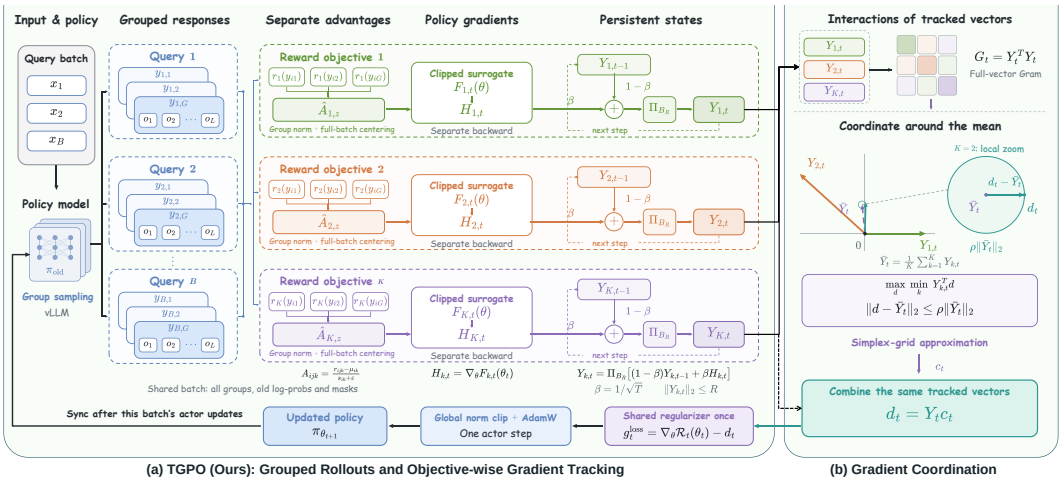

Figure 2: **TGPO method overview: objective identity throughout the update.** (a) Shared rollouts yield separately normalized advantages and clipped policy gradients, each updating its own bounded state. (b) State inner products define the coordination problem; its coefficients combine those same states into the ascent direction. Shared regularization, global clipping, and AdamW complete the actor update, after which the rollout policy is synchronized at the batch boundary.

Here, N_t counts valid tokens in the logical actor minibatch. Objectives share responses, old-policy probabilities, masks, and parameters. Separate differentiation preserves each objective's clipping branch before coordination. $H_{k,t}$ is an ascent gradient of the normalized surrogate, with rewards and advantages held fixed. Appendix B.5 specifies numerical safeguards and microbatch aggregation. Shared regularization enters after reward coordination.

3.2 A PERSISTENT STATE FOR OBJECTIVE DIRECTIONS

TGPO maintains one vector $Y_{k,t} \in \mathbb{R}^P$ per objective, initialized at zero:

$$Y_{k,t} = \Pi_{\mathbb{B}_R}((1-\beta)Y_{k,t-1} + \beta H_{k,t}), \qquad \mathbb{B}_R = \{v : \|v\|_2 \le R\}, \qquad \beta = T^{-1/2}. \tag{4}$$

Here, T is the planned total number of actor optimizer updates. The recursion makes estimation depend on a history of objective directions; projection bounds the stored state without forcing every objective to unit norm. It therefore retains magnitude differences below R. The time scale is shared across objectives and fixed for the run. Current experiments use $R = 1$ and update every tracker at every actor step.

3.3 COORDINATING THE LEAST FAVORABLE PREDICTED IMPROVEMENT

Let $Y_t = [Y_{1,t}, \ldots, Y_{K,t}]$, $a = \mathbf{1}/K$, $\bar{Y}_t = Y_t a$, and $G_t = Y_t^\top Y_t$. TGPO coordinates local progress by the bounded max–min program

$$\max_d \min_k Y_{k,t}^\top d \quad \text{subject to} \quad \left\|d - \bar{Y}_t\right\|_2 \le \rho \left\|\bar{Y}_t\right\|_2, \qquad 0 \le \rho < 1. \tag{5}$$

The mean direction anchors aggregate progress; the radius determines how far coordination can alter it to improve the least favorable tracked margin. The constraint is in gradient space. This local criterion follows the conflict-averse formulation of Liu et al. (2021), applied here to persistent policy-gradient states. Its simplex dual is

$$w_t^* \in \arg\min_{w \in \Delta_K} \left\{ w^\top G_t a + \rho \sqrt{a^\top G_t a} \sqrt{w^\top G_t w} \right\}, \quad \Delta_K = \{w \ge 0 : \mathbf{1}^\top w = 1\}. \tag{6}$$

For $\|Y_t w_t^*\| > 0$, recover

$$d_t = \bar{Y}_t + \rho \frac{\|\bar{Y}_t\|}{\|Y_t w_t^*\|} Y_t w_t^* = Y_t c_t, \qquad c_t = a + \rho \frac{\|\bar{Y}_t\|}{\|Y_t w_t^*\|} w_t^*. \tag{7}$$

The implementation searches a resolution-100 simplex grid for $K \in \{2, 3\}$ and uses $\rho = 0.125$. Norm denominators receive a numerical floor; if $\left\|\bar{Y}_t\right\| \leq 10^{-12}$, the runtime returns $\bar{Y}_t$. Appendix A.4 derives the dual, proves primal recovery, and analyzes degenerate solutions. The returned grid solution is an approximation to the continuous optimum.

3.4 Executing the tracked update

With shared regularizer $\mathcal{R}_t$, the minimizing gradient passed to the trainer is

$$g_t^{\text{loss}} = \nabla \mathcal{R}_t(\theta_t) - d_t, \qquad \theta_{t+1} = \text{AdamW}(\theta_t, \text{Clip}_1(g_t^{\text{loss}})). \tag{8}$$

Here, Clip_1 clips the full gradient to Euclidean norm at most one; the scheduler advances once per outer rollout batch. A scalar combined policy loss is not differentiated again. Figure 2 distinguishes the two uses of Y_t: estimating interactions and executing their coordinated combination. The full algorithm and memory accounting appear in Appendix B.1.

4 Understanding temporal gradient coordination

Proposition 1. Controlled Change To The Aggregate Direction. *Every direction feasible for Eq. (5) satisfies*

$$\bar{Y}_t^\top d \geq (1-\rho)\left\|\bar{Y}_t\right\|^2, \qquad \|d\| \leq (1+\rho)\left\|\bar{Y}_t\right\|. \tag{9}$$

The continuous optimum attains a minimum tracked margin at least as large as that of $\bar{Y}_t$.

Thus coordination preserves positive alignment with a nonzero average while reallocating local progress toward the least favorable objective. The comparison concerns the continuous problem; the implementation's grid approximation preserves the radius constraint through its reconstruction. Appendix A.5 proves both bounds and the minimum-margin comparison.

Tracking Separates Sampling Variation From Target Drift. Let $m_{k,t}$ be a reference mean surrogate gradient inside $\mathbb{B}_R$, and decompose the instantaneous input as $H_{k,t} = m_{k,t} + b_{k,t} + \xi_{k,t}$. Conditional on the history, assume $\mathbb{E}[\xi_{k,t}] = 0$ and $\mathbb{E}\left\|\xi_{k,t}\right\|^2 \leq \nu_{k,t}^2$. For $e_{k,t} = Y_{k,t} - m_{k,t}$ and $\delta_{k,t} = m_{k,t-1} - m_{k,t}$, projection nonexpansiveness gives

$$\mathbb{E}_t\left\|e_{k,t}\right\|^2 \leq \left\|(1-\beta)(e_{k,t-1} + \delta_{k,t}) + \beta b_{k,t}\right\|^2 + \beta^2 \nu_{k,t}^2. \tag{10}$$

This bound exposes the design trade-off: smaller β attenuates fresh sampling variation but retains more drift from earlier policies. In the stationary unbiased case, the asymptotic mean-square bound is $\beta\nu_k^2/(2-\beta)$. Group normalization and changing policies motivate retaining the bias and drift terms. Appendix A.6 derives the recursion and its stationary bound.

Proposition 2. Executed Progress With Estimation Error. *Let F_k have an L_k-Lipschitz gradient along the segment from θ_t to $\theta_t + \Delta_t$, and let $\|Y_{k,t} - \nabla F_k(\theta_t)\| \leq \varepsilon_{k,t}$. Then*

$$F_k(\theta_t + \Delta_t) - F_k(\theta_t) \geq Y_{k,t}^\top \Delta_t - \varepsilon_{k,t}\left\|\Delta_t\right\| - \tfrac{L_k}{2}\left\|\Delta_t\right\|^2. \tag{11}$$

The bound identifies three quantities that determine local progress: the executed tracked margin, estimation error, and curvature cost. It applies to a smooth surrogate region; clipped objectives are piecewise smooth. Using the actual Δ_t also accounts for AdamW preconditioning, momentum, regularization, and weight decay. Appendix A.7 proves the bound and derives a sufficient step-size condition for simultaneous surrogate improvement.

Why The Vector State Matters. Without active projection, $Y_{k,t} = \beta\sum_{s=1}^{t}(1-\beta)^{t-s}H_{k,s}$. Thus d_t can retain directions outside the span of the current gradients. Updating with current gradients and coefficients inferred from a smoothed Gram matrix is generally a different operation. Appendix A.8 constructs a two-dimensional example proving this distinction; Appendix A.2 separately shows why reward statistics cannot identify gradient alignment.

5 EXPERIMENTS

5.1 TASKS, COMPARISONS, AND EVALUATION

We compare TGPO (Ours), GDPO, and DVAO on mathematics, code generation, and tool use with GDPO's reward designs (Liu et al., 2026a; Jiang et al., 2026a). Mathematics uses DeepSeek-R1-Distill-Qwen at 1.5B/7B, code its 1.5B model, and tools Qwen2.5-Instruct at 1.5B/3B (Guo et al., 2025; Qwen Team, 2025). We update all parameters for 500, 400, and 100 rollout steps on DeepScaleR, Eurus-2-RL, and the 3,920-example RLLA split, respectively (Luo et al., 2025; Cui et al., 2026; Qian et al., 2025). Appendix B gives rewards, settings, and prompt formats.

Mathematical evaluation covers MATH, AIME24, AMC22/23, Minerva, and Olympiad (Hendrycks et al., 2021b; Lewkowycz et al., 2022; He et al., 2024); macro scores weight suites equally. Code evaluation covers APPS, CodeContests, Codeforces, and TACO (Hendrycks et al., 2021a; Li et al., 2022; 2023). Both reasoning tasks use four samples per question. Tool evaluation retains all 4,441 BFCL-v3 items (Patil et al., 2025). Training curves and held-out endpoint scores are reported separately.

5.2 MATHEMATICAL REASONING ACROSS MODEL SCALES

As shown in Table 1, TGPO attains the best joint accuracy–length outcome in the 1.5B comparison: macro accuracy is 33.07%, versus 30.74% for GDPO and 29.65% for DVAO, while overlength is 0.40%, versus 1.28% and 2.24%. Accuracy improves on four of five suites. Figure 3 shows earlier length control and stronger subsequent correctness learning.

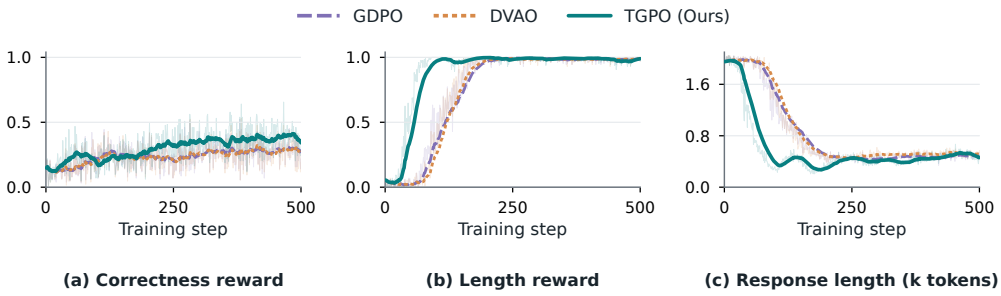

Figure 3: 1.5B mathematical training: correctness and length rewards, and mean response length.

At 7B, TGPO leads on MATH500, Minerva, and Olympiad and reduces macro overlength to 1.33%, versus 5.81% and 4.30%. Its macro accuracy is 42.65%, versus 44.72% and 44.02%; question-weighted accuracy reaches 50.67%, versus 50.08% and 49.73%, as detailed in Table 6 in Appendix C.

Figure 8 shows the three final 1.5B mathematical checkpoints in accuracy–length space. At $(33.07\%, 99.60\%)$, TGPO improves both coordinates over both baselines and is the only non-dominated configuration.

Figure 4 demonstrates how correctness and brevity coincide in a matched 7B AMC22/23 response. GDPO treats diagonal BD as a side and obtains a coordinate inconsistent with the perpendicular. DVAO leaves the known side $AB = AD = 5$ unused, then substitutes a base segment for the height. TGPO uses the side immediately, derives $BP = 4$, and obtains area 20 in 397 tokens; the incorrect baseline responses use 766 and 784 tokens. Carrying this geometric constraint through each step yields a shorter solution.

5.3 CODE GENERATION UNDER LENGTH CONSTRAINTS

Figure 5 shows that TGPO improves all three late-training reward means. As reported in Table 2, on held-out questions, correct responses within 4,000 tokens reach 15.73%, versus 13.54% for GDPO and 13.81% for DVAO. Overlength falls to 8.53% and mean output length to 2,666 tokens. Unrestricted pass@1 is 15.83%, versus 16.49% and 16.72%.

Table 1: Mathematical reasoning after 500 updates (%). Overlength exceeds 1,024 tokens; the generation cap is 2,048. Macro averages weight all five benchmarks equally. Bold marks the best result within each scale. Baselines are GDPO (Liu et al., 2026a) and DVAO (Jiang et al., 2026a).

	Accuracy ↑			Overlength ↓		
Benchmark	GDPO	DVAO	TGPO (Ours)	GDPO	DVAO	TGPO (Ours)
DeepSeek-R1-Distill-Qwen-1.5B (Guo et al., 2025)						
MATH500 (Lightman et al., 2023)	64.30	64.55	**67.75**	0.70	0.80	**0.05**
AIME24 (MAA, n.d.a)	**9.17**	5.00	3.33	3.33	5.83	**0.83**
AMC22/23 (MAA, n.d.b)	35.24	34.64	**45.48**	**0.30**	1.81	0.60
Minerva (Lewkowycz et al., 2022)	19.49	18.66	**19.94**	0.28	0.18	**0.09**
Olympiad (He et al., 2024)	25.52	25.41	**28.85**	1.78	2.59	**0.41**
Macro average	30.74	29.65	**33.07**	1.28	2.24	**0.40**
DeepSeek-R1-Distill-Qwen-7B (Guo et al., 2025)						
MATH500 (Lightman et al., 2023)	78.05	78.85	**79.45**	2.05	1.90	**1.30**
AIME24 (MAA, n.d.a)	22.50	**24.17**	15.00	17.50	10.00	**0.00**
AMC22/23 (MAA, n.d.b)	**57.23**	52.41	50.60	6.02	4.82	**3.01**
Minerva (Lewkowycz et al., 2022)	26.65	26.29	**28.22**	**0.18**	1.29	1.10
Olympiad (He et al., 2024)	39.15	38.41	**40.00**	3.30	3.52	**1.26**
Macro average	**44.72**	44.02	42.65	5.81	4.30	**1.33**

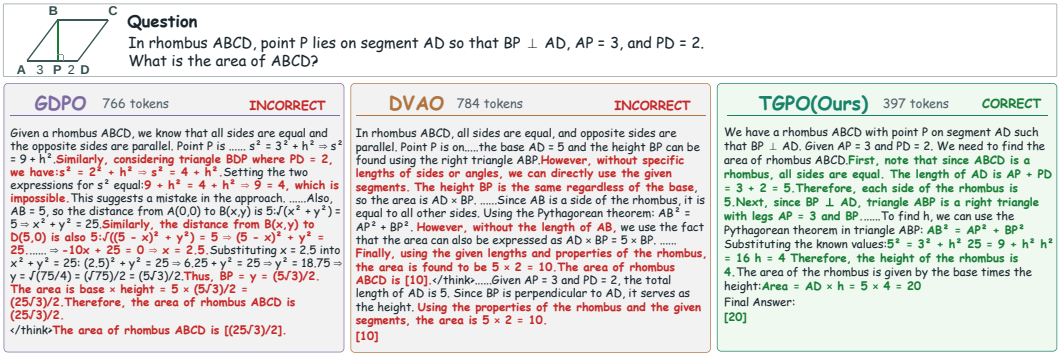

Figure 4: **Reasoning demonstration: the same geometry problem, three solution paths.** Red highlights the decisive errors in GDPO and DVAO: confusing a diagonal with a side and substituting a base segment for the height. Green traces TGPO's consistent use of the side length to derive the height and area. Token counts refer to the complete recorded responses.

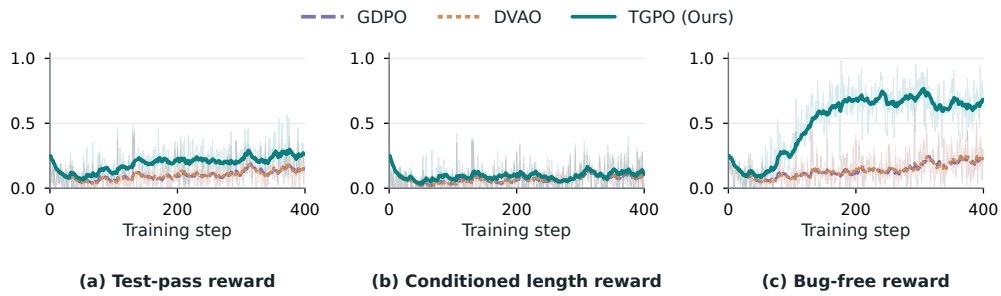

Figure 5: 1.5B code training across all three optimized rewards. The test-pass reward measures the fraction of tests passed.

Table 2: Code generation with DeepSeek-R1-Distill-Qwen-1.5B (Guo et al., 2025) on APPS (Hendrycks et al., 2021a), CodeContests (Li et al., 2022), Codeforces (MatrixStudio, n.d.), and TACO (Li et al., 2023): 1,014 questions, four samples each. Rates are percentages; generation is capped at 32,768 tokens and overlength exceeds 4,000. Baselines are GDPO (Liu et al., 2026a) and DVAO (Jiang et al., 2026a).

Metric	GDPO	DVAO	TGPO (Ours)
Pass@1 ↑	16.49	**16.72**	15.83
Correct and ≤4,000 tokens ↑	13.54	13.81	**15.73**
Overlength ↓	62.97	61.98	**8.53**
Syntax-valid ↑	93.69	95.07	**98.77**
Mean tokens ↓	7298.08	6885.39	**2666.17**

5.4 Tool use and structural adherence

Table 3 shows that TGPO achieves the highest held-out strict-format and multi-turn trajectory-format rates at both scales. At 1.5B, overall accuracy is 40.09%, versus 37.46% for GDPO. At 3B, strict format reaches 99.07% and multi-turn format 89.38%, versus GDPO's 97.66% and 75.25%.

Table 3: BFCL-v3 (Patil et al., 2025) using Qwen2.5-Instruct-1.5B/3B (Qwen Team, 2025) (%). Accuracy uses official aggregation; strict format is response-weighted; multi-turn format requires an entirely compliant trajectory. All 4,441 questions are retained. Baselines are GDPO (Liu et al., 2026a) and DVAO (Jiang et al., 2026a).

	Qwen2.5-Instruct-1.5B			Qwen2.5-Instruct-3B		
Endpoint ↑	GDPO	DVAO	TGPO (Ours)	GDPO	DVAO	TGPO (Ours)
Overall accuracy	37.46	**44.00**	40.09	**51.59**	50.92	47.28
Strict format	92.73	77.18	**96.06**	97.66	94.18	**99.07**
Multi-turn format	58.38	16.63	**77.88**	75.25	54.75	**89.38**

The mechanism study in Appendix C.2 compares the complete method with each component removed under a common BFCL-v3 protocol. Both tracking and geometric coordination contribute to the structural endpoints.

Joint Reward Evolution. Figure 6 presents the four recorded mathematical training signals over the same 500 steps. Total reward reflects the combined correctness and length objectives, while format is monitored as a diagnostic. TGPO establishes length compliance early and continues improving correctness, supporting a common update that preserves the contributions of both objectives as learning progresses.

5.5 How tracking and coordination parameters shape the update

We examine $\beta \in \{1/\sqrt{500}, 0.1\}$ and $\rho \in \{0.0625, 0.125\}$ in four 1.5B runs initialized from the same base model, each with 12 actor updates. To isolate geometric effects, we also recompute directions at fixed tracker states from the 500-step mathematical training trajectory, as shown in Figure 7.

Directional Memory. With inactive projection, a gradient's relative weight after j updates is $(1-\beta)^j$. Increasing β from $1/\sqrt{500}$ to 0.1 shortens the weight half-life from 15.15 to 6.58 updates. The smaller rate integrates longer directional histories; the larger rate emphasizes recent gradients.

Bounded Geometric Adjustment. Halving ρ reduces the permitted correction from 12.5% to 6.25% of the mean-direction norm. Across the 500 fixed states, the median angular change is 3.016°, with all direction cosines above 0.9982. The radius therefore provides fine control over objective

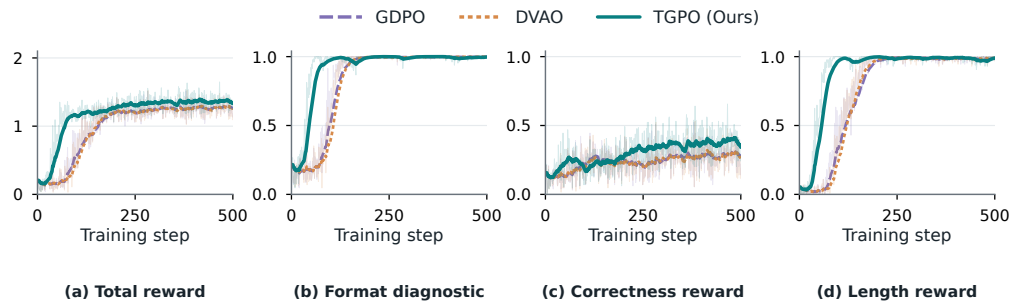

Figure 6: Mathematical training over 500 steps. Panels show total reward, format diagnostic, correctness reward, and length reward. Faint curves retain the recorded values; darker curves show an exponential moving average.

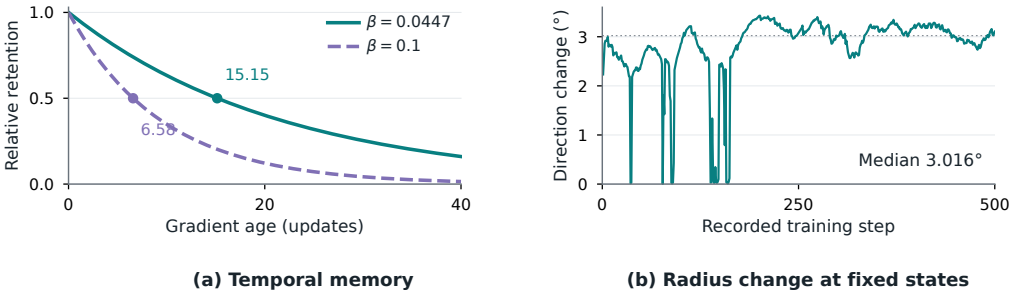

Figure 7: **Two controls on the coordinated update.** (a) Analytical retention of a past gradient under the two tracking rates; dots mark the weight half-lives. (b) Directional change when ρ is halved from 0.125 to 0.0625 at each of 500 recorded tracker states. The temporal kernel determines which directions are retained, while the radius controls their local geometric adjustment.

balancing while preserving a common ascent anchor. At $\rho = 0$, coordination reduces to the mean; increasing ρ expands the feasible set for improving the least favorable tracked margin.

Coupling Through Tracked Geometry. Changing β reweights past gradients and hence reshapes $G_t = Y_t^\top Y_t$; changing ρ adjusts the coordination problem defined by that geometry. Common positive rescaling leaves the dual weights and combination coefficients unchanged when numerical floors are inactive. Thus β influences geometry through directional history, while ρ controls the permitted adjustment around the resulting mean.

6 Conclusion

TGPO unifies reward decoupling, temporal gradient estimation, and geometric coordination in multi-reward policy optimization. A separate gradient state for each objective carries directional history into both coordination and the policy update. The mean direction provides a common reference, while a bounded correction redistributes local progress toward the least favorable tracked objective. This compact optimization connects objective-specific information to a shared update.

Evaluations against GDPO and DVAO on mathematical reasoning, code generation, and tool use demonstrate stronger aggregate constraint adherence across tasks and model scales, combining correctness with effective control of response length and structure. Theory connects mean alignment and estimation error to executed local progress. Ablations and parameter analysis clarify the complementary roles of temporal memory and geometric coordination. Together, these findings establish persistent objective directions as an effective basis for multi-reward learning, with natural extensions to richer reward sets and adaptive tracking schedules.

Ethics statement

This study uses existing mathematical, programming, and tool-use benchmarks, with no human participants or new personal-data collection. Models and datasets are used subject to their respective licenses; generated code is evaluated in an isolated environment.

Reproducibility statement

Section 3 and Appendices B.1–C specify the algorithm, training and evaluation protocols, and extended results.

AI use statement

AI tools were used to polish the manuscript.

References

Pranjal Aggarwal and Sean Welleck. L1: Controlling how long a reasoning model thinks with reinforcement learning. In *Second Conference on Language Modeling*, 2025.

Yi Chen, Yuying Ge, Rui Wang, Yixiao Ge, Junhao Cheng, Ying Shan, and Xihui Liu. Grpo-care: Consistency-aware reinforcement learning for multimodal reasoning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 4305–4320, 2026.

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. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856.

Heshan Devaka Fernando, Han Shen, Miao Liu, Subhajit Chaudhury, Keerthiram Murugesan, and Tianyi Chen. Mitigating gradient bias in multi-objective learning: A provably convergent approach. In *The eleventh international conference on learning representations*, 2023.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen 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. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, 2024.

Dan Hendrycks, Steven Basart, Saurav Kadavath, Mantas Mazeika, Akul Arora, Ethan Guo, Collin Burns, Samir Puranik, Horace He, Dawn Song, and Jacob Steinhardt. Measuring coding challenge competence with APPS. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021a.

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. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021b.

Guochao Jiang, Jingyi Song, Guofeng Quan, Chuzhan Hao, Guohua Liu, and Yuewei Zhang. DVAO: Dynamic variance-adaptive advantage optimization for multi-reward reinforcement learning. *arXiv preprint arXiv:2605.25604*, 2026a.

Shuyang Jiang, Yusheng Liao, Ya Zhang, Yanfeng Wang, and Yu Wang. Overthinking reduction with decoupled rewards and curriculum data scheduling. In *International Conference on Learning Representations*, volume 2026, pp. 114966–114995, 2026b.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 3843–3857. Curran Associates, Inc., 2022. doi: 10.52202/068431-0278.

Rongao Li, Jie Fu, Bo-Wen Zhang, Tao Huang, Zhihong Sun, Chen Lyu, Guang Liu, Zhi Jin, and Ge Li. Taco: Topics in algorithmic code generation dataset. *arXiv preprint arXiv:2312.14852*, 2023.

Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, et al. Competition-level code generation with alphacode. *Science*, 378(6624):1092–1097, 2022.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 18878–18890. Curran Associates, Inc., 2021.

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. In *Forty-third International Conference on Machine Learning*, 2026a.

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. In *International Conference on Learning Representations*, volume 2026, pp. 43279–43308, 2026b.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Erran Li, Raluca Ada Popa, and Ion Stoica. DeepScaleR: Surpassing O1-Preview with a 1.5B model by scaling RL. Project report, 2025.

MAA. American invitational mathematics examination. Mathematical Association of America, n.d.a. Accessed September 25, 2026.

MAA. American mathematics competitions. Mathematical Association of America, n.d.b. Accessed September 25, 2026.

MatrixStudio. Codeforces-Python-Submissions. Hugging Face dataset, n.d. Accessed September 25, 2026.

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. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 108877–108901. Curran Associates, Inc., 2024. doi: 10.52202/079017-3457.

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.

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru WANG, Xiusi Chen, Dilek Hakkani-Tur, Gokhan Tur, and Heng Ji. Toolrl: Reward is all tool learning needs. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 105523–105553. Curran Associates, Inc., 2025. doi: 10.52202/085713-3524.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025.

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

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

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*, pp. 1279–1297, 2025.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, juncai liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 113222–113244. Curran Associates, Inc., 2025. doi: 10.52202/085713-3775.

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

A Mathematical details and proofs

A.1 Objects of the analysis

We distinguish the raw reward objective J_k, the batch-defined normalized policy surrogate $F_{k,t}$, its instantaneous gradient $H_{k,t}$, the tracked vector $Y_{k,t}$, and the actual optimizer displacement $\Delta_t = \theta_{t+1} - \theta_t$. A statement about one of these objects does not automatically transfer to the others. In particular, the data-dependent group standard deviation changes the weighting of the score-function estimator, clipping is nonlinear in the advantage, and AdamW transforms the combined gradient. The analysis therefore states its target and assumptions explicitly at each step.

A.2 Reward statistics do not identify score geometry

Consider three equally likely actions at $\theta = 0$ and two centered rewards $A_1 = (1, -1, 0)$ and $A_2 = (0, 1, -1)$. A scalar-parameter softmax policy with logits $s_i\theta$ has score s_i at zero whenever $\sum_i s_i = 0$. For $s = (1, 0, -1)$, the two reward gradients are $g_1 = g_2 = 1/3$, so their product is positive. For $s = (0, 1, -1)$, they are $g_1 = -1/3$ and $g_2 = 2/3$, so their product is negative. The action probabilities, reward values, reward variances, and reward covariance are identical in both cases. Positive group rescaling preserves the sign difference. Thus those reward statistics cannot identify gradient alignment without information about the policy score.

This example concerns distinguishability, not an empirical comparison. It also clarifies why taking derivatives of a scalar advantage with respect to reward values does not substitute for computing parameter-gradient interactions: the policy Jacobian remains between rewards and parameter changes.

A.3 Scalar aggregation and the clipping boundary

For group size $G > 1$, define $\mu_{ik} = G^{-1} \sum_j r_{ijk}$, $s_{ik}^2 = (G-1)^{-1} \sum_j (r_{ijk} - \mu_{ik})^2$, and $\sigma_{ik}^2 = G^{-1} \sum_j (r_{ijk} - \mu_{ik})^2$. GDPO and TGPO use s_{ik} with additive $\epsilon = 10^{-6}$. The local DVAO baseline uses σ_{ik} after its reward-bound preprocessing. Its implemented advantage is

$$A_{ij}^{\mathrm{DVAO}} = \begin{cases} \dfrac{\sum_k b_k (r_{ijk} - \mu_{ik})}{\sum_k b_k \sigma_{ik}}, & \sum_k b_k \sigma_{ik} > \epsilon, \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

For positive group variance, this equals a variance-adaptive mixture of unregularized component advantages: $\sum_k \widetilde{b}_{ik} (r_{ijk} - \mu_{ik})/\sigma_{ik}$, with zero-variance components contributing zero. It does not follow by cancelling a denominator $s_{ik} + \epsilon$; the numerical conventions are different. Figure 1 shows the implemented centered-reward form. The fixed b_k are base preferences, whereas the effective DVAO weights $\widetilde{b}_{ik}$ vary by group.

Scalar aggregation and separate clipping also represent distinct operations. In a common unclipped linear regime, differentiation commutes with a fixed weighted sum. Across clipping branches, in general,

$$\min\{q \textstyle\sum_k b_k A_k, \mathrm{clip}(q) \sum_k b_k A_k\} \neq \sum_k b_k \min\{q A_k, \mathrm{clip}(q) A_k\}. \tag{13}$$

For example, take equal weights, $A_1 = 1$, $A_2 = -1$, $q = 1.5$, and upper clipping threshold 1.2. The left side is zero; the right side is -0.15. TGPO therefore defines its separate clipped surrogates directly in Eq. (3).

A.4 Deriving the coordination dual

Write $b = Ya$ and $r = \rho \|b\|$. Because the minimum of a finite collection equals the minimum over its convex combinations,

$$\max_{\|d-b\| \le r} \min_k Y_k^\top d = \max_{\|u\| \le r} \min_{w \in \Delta_K} (Yw)^\top (b + u) \tag{14}$$

$$= \min_{w \in \Delta_K} \max_{\|u\| \le r} (Yw)^\top (b + u) \tag{15}$$

$$= \min_{w \in \Delta_K} \{w^\top G a + r\sqrt{w^\top G w}\}. \tag{16}$$

The interchange follows from compact convex domains and a bilinear objective. For an optimal w^* with $Yw^* \neq 0$, the support function of the Euclidean ball yields $u^* = rYw^*/\|Yw^*\|$, giving Eq. (7). To verify primal recovery, differentiate the dual at such a solution: $\nabla\phi(w^*) = Y^\top d^*$. The simplex optimality condition is $(w - w^*)^\top Y^\top d^* \geq 0$ for every $w \in \Delta_K$. Hence w^* minimizes $(Yw)^\top d^*$, so $\min_k Y_k^\top d^* = (Yw^*)^\top d^* = \phi(w^*)$. Feasibility and equality to the dual value prove primal optimality. When $b = 0$, the feasible set is the singleton $d = 0$. When $b \neq 0$ but $Yw^* = 0$, the dual norm is nonsmooth: an exact primal reconstruction requires a compatible subgradient satisfying the simplex conditions. The runtime instead uses a denominator floor of 10^{-12}; this always bounds the adjustment norm, but is not an exact solver for all degenerate continuous cases.

Finite-Grid Objective Error. If $\|Y_k\| \leq R$, the dual objective $\phi(w)$ is $(1+\rho)R\,\|b\|$-Lipschitz in ℓ_1. A resolution-M simplex grid admits, for every w, a grid point within $2(K-1)/M$ in ℓ_1: round the first $K-1$ entries down and assign the remainder to the last entry. Therefore

$$0 \leq \min_{w \in \Delta_K^{(M)}} \phi(w) - \min_{w \in \Delta_K} \phi(w) \leq \frac{2(K-1)(1+\rho)R\,\|b\|}{M}. \tag{17}$$

This is a bound on the dual objective approximation, not on the reconstructed primal margin near a singularity. The experiments use $M = 100$, giving 101 candidates for two objectives and 5,151 for three.

A.5 Proof of Proposition 1

Proof. Write $d = b + u$ with $\|u\| \leq \rho\,\|b\|$. Cauchy–Schwarz gives $b^\top d = \|b\|^2 + b^\top u \geq (1-\rho)\,\|b\|^2$, and the triangle inequality gives $\|d\| \leq (1+\rho)\,\|b\|$. Since b is feasible, the optimum of the minimum-margin objective is at least its value at b. The numerical reconstruction also has $\|u\| \leq \rho\,\|b\|$ because flooring the denominator can only decrease the norm of its correction. □

Positive mean alignment does not imply that every tracked margin is positive. If 0 lies in the convex hull of the objective gradients, no direction can have a strictly positive inner product with all of them: taking the same convex combination of those positive inner products would contradict its value of zero. This geometric fact explains why a coordination algorithm still expresses a trade-off when objectives are locally incompatible.

A.6 Tracking-error recursion and its stationary case

Condition on the history before observing $H_{k,t}$. Assume the reference $m_{k,t} \in \mathbb{B}_R$, drift $\delta_{k,t}$, and bias $b_{k,t}$ are measurable with respect to that history. Since $\Pi_{\mathbb{B}_R}(m_{k,t}) = m_{k,t}$, nonexpansiveness gives

$$\|Y_{k,t} - m_{k,t}\|^2 \leq \|(1-\beta)(Y_{k,t-1} - m_{k,t}) + \beta b_{k,t} + \beta\xi_{k,t}\|^2 \tag{18}$$

$$= \|(1-\beta)(e_{k,t-1} + \delta_{k,t}) + \beta b_{k,t} + \beta\xi_{k,t}\|^2. \tag{19}$$

Taking conditional expectation eliminates the cross term with $\xi_{k,t}$ and proves Eq. (10). In the stationary unbiased case with a uniform noise bound ν_k^2,

$$\mathbb{E}\,\|e_{k,t}\|^2 \leq (1-\beta)^{2t}\,\|e_{k,0}\|^2 + \frac{\beta}{2-\beta}\left[1 - (1-\beta)^{2t}\right]\nu_k^2. \tag{20}$$

The general inequality retains drift and bias because both can matter in policy optimization. In particular, this stationary calculation is an explanatory special case and not a claim that the evolving language-model gradients satisfy stationarity.

A.7 Proof of Proposition 2

Proof. The smoothness lower bound gives $F_k(\theta + \Delta) - F_k(\theta) \geq \nabla F_k(\theta)^\top \Delta - L_k\,\|\Delta\|^2/2$. Substituting $\nabla F_k(\theta) = Y_k + (\nabla F_k(\theta) - Y_k)$ and applying Cauchy–Schwarz to the error term proves the result. □

For a direct step $\Delta = \eta d$, define $\gamma = \min_k Y_k^\top d$, $\varepsilon = \max_k \varepsilon_k$, and $L = \max_k L_k$. If $\gamma > \varepsilon \|d\|$ and $L > 0$, every analyzed surrogate increases whenever

$$0 < \eta < \frac{2(\gamma - \varepsilon \|d\|)}{L \|d\|^2}. \tag{21}$$

With AdamW, the relevant quantity is instead $Y_k^\top \Delta$ for the executed displacement. The same proposition applies when its smoothness and error assumptions hold. This distinction motivates logging both pre-optimizer and executed margins in mechanism diagnostics. It does not require changing the baseline optimizer.

A.8 A VECTOR-MEMORY EXAMPLE

Suppose both current objective gradients are $H_{1,t} = H_{2,t} = e_1$, and both previous states are $Y_{1,t-1} = Y_{2,t-1} = e_2$. For $0 < \beta < 1$ and $R \geq 1$, projection is inactive and each new state is $\beta e_1 + (1-\beta)e_2$. Any weighted combination of current gradients lies in $\mathrm{span}\{e_1\}$. TGPO's combination has a nonzero e_2 component because its effective coefficients sum to a positive value. Thus no scalar weights on the current gradients can reproduce this update. The example demonstrates why storing a Gram matrix or smoothing weights is insufficient to preserve the semantics of full-vector tracking.

A.9 A REPRODUCIBLE CONSTRUCTION FOR THE MOTIVATION FIGURE

The geometric margins in Figure 1 can be realized by the following two tracked vectors, with $R = 1$ and $\rho = 1/8$:

$$\begin{aligned} &Y_1 = (0.6, 0)^\top, \qquad Y_2 = (-0.66, 0.6)^\top, \\ &\bar{Y} = (-0.03, 0.3)^\top, \qquad G = \begin{pmatrix} 0.36 & -0.396 \\ -0.396 & 0.7956 \end{pmatrix}. \end{aligned} \tag{22}$$

Both vectors satisfy the norm bound. Let $r = \left\|\bar{Y}\right\|/8 = \sqrt{0.0909}/8$ and $d^* = \bar{Y} + (r, 0)^\top$. For every feasible d, Cauchy–Schwarz gives $\min_k Y_k^\top d \leq Y_1^\top d \leq Y_1^\top \bar{Y} + r \|Y_1\|$. The candidate d^* attains the last bound, with

$$Y_1^\top \bar{Y} = -0.0180, \qquad Y_1^\top d^* = -0.018 + 0.6\sqrt{0.0909}/8 = 0.0046122201\ldots, \tag{23}$$

while $Y_2^\top d^* = 0.1749265578\ldots > Y_1^\top d^*$. Hence d^* solves the primal problem, and $w^* = (1, 0)^\top$ solves its dual. This simplex vertex is also present in the implemented grid. Rounding to four decimals gives the two displayed margins. The dashed boundary is $Y_1^\top d = 0$; the mean lies on its negative side and the coordinated direction on its positive side. The drawing is schematic and its search region is magnified. These numbers describe tracked first-order geometry, independently of the training curves and of the later AdamW displacement.

B ALGORITHM AND EXPERIMENTAL PROTOCOLS

B.1 COMPLETE UPDATE PROCEDURE

Algorithm 1 Tracked-Gradient Policy Optimization

Require: Initial policy θ_1; K reward functions; T actor updates; $R = 1$; $\rho = 0.125$; $M = 100$

Initialize $Y_{k,0} = 0$ for each objective; set $\beta = 1/\sqrt{T}$ and $t = 1$
for each rollout batch **do**
  Sample grouped responses and evaluate all reward components
  Normalize each objective within prompt groups; center over the complete accepted batch
  for each actor minibatch **do**
    for $k = 1, \ldots, K$ **do**
      Compute $H_{k,t} = \nabla F_{k,t}(\theta_t)$ with separate clipped surrogate
      $Y_{k,t} \leftarrow \Pi_{\mathbb{B}_R}((1-\beta)Y_{k,t-1} + \beta H_{k,t})$
    end for
    Reduce full-parameter inner products to obtain $G_t = Y_t^\top Y_t$
    Minimize Eq. (6) on $\Delta_K^{(M)}$ and reconstruct d_t
    Load $\nabla\mathcal{R}_t - d_t$, clip its global norm, and take one AdamW step
    Increment the actor-update counter $t \leftarrow t + 1$
  end for
  Advance the scheduler once; synchronize the rollout policy
end for

All objectives in an actor minibatch use the same stochastic-mask state. Shared entropy or explicit KL gradients are evaluated once after the objective passes. A zero instantaneous gradient decays the historical state; it does not reset it. Checkpointing a continuing run requires the tracker states as well as policy, optimizer, scheduler, random-number, and data-cursor states.

B.2 DISTRIBUTED REPRESENTATION

The tracker stores FP32 parameter shards on CPU, one vector per objective. Global norms and the $K \times K$ Gram matrix are accumulated in FP64 and reduced across ranks. Direction assembly is chunked before the clipped optimizer step. Storage is $O(KP)$ for P trainable parameters, and forming the Gram matrix costs $O(K^2P)$ arithmetic.

B.3 REWARD DEFINITIONS

Mathematics. Correctness is binary boxed-answer agreement under the common DeepScaleR grader. Length reward is $\mathbb{I}[|y| \leq 1024]$. These are two separate optimized objectives; the recorded format score is a diagnostic. Neither reward is replaced by the held-out benchmark score.

Code. The training rewards are test-case pass fraction, $\mathbb{I}[\text{all tests pass}]\mathbb{I}[|y| \leq 4000]$, and the retained bug-free indicator. The PRIME-derived training grader includes its existing first-ten-case fallback. Held-out strict pass@1 instead requires passing every test associated with the evaluated question. We therefore report it as an all-tests success rate, not a mean fraction of passing tests. Syntax-valid output is an additional diagnostic and is not identified with the training bug-free reward.

Tools. The retained ToolRL implementation scores call correctness in $[-3, 3]$ and format in $[0, 1]$. The local DVAO baseline maps correctness to its bounded reward convention and uses equal base weights. The tool trainer logs a reward-side KL diagnostic, but the decoupled objective advantages are constructed from the separate correctness and format components. The active TGPO update uses entropy regularization with coefficient 0.001 and no explicit actor KL loss.

B.4 TRAINING SETTINGS AND MODEL-SCALE EXTENSIONS

Tables 4 and 5 summarize the training budgets and optimization settings.

Table 4: Training budgets for all completed comparisons. Task designs follow GDPO (Liu et al., 2026a).

Task	Model size	K	Updates	Batch	Group	Cap
Mathematics	1.5B / 7B	2	500	16	8	2,048
Code	1.5B	3	400	8	8	5,120
Tool use	1.5B / 3B	2	400	512	4	1,024

Updates count actor steps; batch counts prompts per rollout; cap is response tokens. Tool training has 100 rollout batches and four actor updates per batch.

Table 5: Training hyperparameters shared by both mathematics scales and both tool scales; code uses 1.5B. The dash in the tool length field means not applicable. Tracking and coordination settings apply to TGPO (Ours). Task designs follow GDPO (Liu et al., 2026a).

Hyperparameter	Mathematics	Code	Tool use
Rollout steps	500	400	100
Actor updates	500	400	400
Prompts per rollout	16	8	512
Responses per prompt	8	8	4
Actor minibatch (prompts)	16	8	128
Prompt cap (tokens)	1,024	1,024	2,048
Response cap (tokens)	2,048	5,120	1,024
Length threshold (tokens)	1,024	4,000	—
Learning rate	10^{-6}	10^{-6}	10^{-6}
PPO epochs	1	1	1
Clip (lower / upper)	0.20 / 0.28	0.20 / 0.28	0.20 / 0.20
Explicit KL coefficient	0.0005	0.0005	0
Logged reward KL coefficient	0	0	0.001
Entropy coefficient	0	0	0.001
Gradient norm cap	1	1	1
Tracking rate	$1/\sqrt{500}$	$1/\sqrt{400}$	$1/\sqrt{400}$
Tracker radius R	1	1	1
Coordination radius ρ	0.125	0.125	0.125
Simplex grid resolution	100	100	100

Mathematics and code use fixed prompt budgets without dynamic group refill. Mathematics processes $500 \times 16 \times 8 = 64{,}000$ generated responses per run; code processes $400 \times 8 \times 8 = 25{,}600$. Tool training uses $100 \times 512 \times 4 = 204{,}800$ responses and four actor minibatches per rollout batch, giving 400 actor updates. Consequently, the tracking horizon is 500 for mathematics and 400 for code and tools. All runs use learning rate 10^{-6}, one PPO epoch, and global gradient norm bound 1.0.

The 7B mathematics experiments use the same rewards, data, logical batch, response cap, and update budget as the 1.5B experiments.

Both tool model sizes start from Qwen2.5-Instruct checkpoints and train for 100 rollout batches. Each run completes 400 actor updates. Results use the final scheduled checkpoint.

B.5 Batch centering, numerical ratios, and loss aggregation

Let $\mathcal{B}$ contain all valid response tokens in an accepted rollout batch. For token z belonging to response y_{ij}, define

$$\widehat{A}_{k,z} = A_{ijk} - \frac{1}{|\mathcal{B}|} \sum_{z' \in \mathcal{B}} A_{i(z')j(z')k}. \tag{24}$$

The mean is computed before dividing the batch into actor minibatches. Reward values, group statistics, advantages, and old-policy log probabilities are detached during differentiation. A constant-reward group has zero centered group advantage. The accepted-batch centering then uses the above common token-weighted mean; it is not repeated independently on physical microbatches.

For numerical stability, the implemented ratio is

$$q_z^{\text{num}}(\theta) = \exp(\text{clip}(\log \pi_\theta(y_z \mid x, y_{<z}) - \log \pi_{\text{old}}(y_z \mid x, y_{<z}), -20, 20)) . \tag{25}$$

Equation (3) uses this ratio in the runtime. It agrees with the mathematical ratio in the interior of the numerical bounds and has the derivative of the bounded expression outside them. Objective clipping and numerical clipping define the piecewise-smooth regions relevant to Proposition 2.

TGPO averages losses over all valid tokens in the logical actor minibatch. For chunks q with n_q tokens and mean surrogate $\bar{F}_q$, the aggregate is $\sum_q n_q \bar{F}_q / \sum_q n_q$. At both tool-model sizes, TGPO uses this token mean and the bounded log ratio, while GDPO and DVAO use weighted microbatch means and unbounded log ratios. Mathematics and code use global token means for all compared methods.

B.6 Mathematical evaluation

The five suites contain 500, 30, 83, 272, and 675 questions, respectively, totaling 1,560. Each question receives four generations with seeds 0–3, temperature 0.6, top-p 0.95, and a 2,048-token cap. Accuracy is the mean of the four binary outcomes per question; the overlength indicator uses the 1,024-token boundary. Each suite is averaged over questions, and the primary aggregate weights suites equally. The supplementary micro-average weights all 1,560 questions equally.

B.7 Code evaluation

The available validation file has 1,016 rows following prior prompt-length filtering. Two shared malformed-data records are excluded before calculating success rates, leaving 1,014 scorable records and 4,056 responses per method. Sampling uses four seeds, temperature 0.6, top-p 0.95, BF16 inference, and a 32,768-token generation cap. Every test has a five-second limit. Judge subprocesses use an 8-GiB virtual-address-space limit. Invalid code, model errors, length truncation, timeouts, and memory-limit failures remain unsuccessful responses.

The primary endpoint is average pass@1 across all four responses.

B.8 Tool evaluation

Table 3 reports three metrics on all 4,441 BFCL-v3 questions, including 800 multi-turn questions. Overall accuracy follows the official arithmetic mean of the non-live, live, and multi-turn scores. Strict format is the fraction of responses satisfying the required structure. Multi-turn format is the fraction of trajectories with at least one emitted response and every response satisfying that structure.

Both model sizes use the final step-100 checkpoint, BF16 inference, and temperature 0.001. Following GDPO's task protocol, evaluation uses the same ToolRL-style prompt, parser, and official scorer.

B.9 Prompt formats

Following GDPO's presentation of task prompts (Liu et al., 2026a), the templates below show the message roles and user-facing fields. Braced fields stand for the original dataset text; the model tokenizer applies its chat template before generation.

Mathematical Reasoning. The DeepScaleR problem and instruction form one user message, following the DeepSeek-R1 prompt format.

```
User:
{mathematical problem} Let's think step by step
and output the final answer within \boxed{}.

Assistant:
{reasoning and boxed final answer}
```

Code Generation. The Eurus-2-RL system message and the programming problem are preserved as separate messages. The user message retains the dataset's Python output instruction.

```
System:
When tackling complex reasoning tasks, you
have access to the following actions:
[ASSESS] [ADVANCE] [VERIFY] [SIMPLIFY]
[SYNTHESIZE] [PIVOT] [OUTPUT]
{original action-format instruction}

User:
{programming problem, input/output format,
 constraints, and examples}
Write Python code to solve the problem.
Present the code in
```python
Your code
```
at the end.

Assistant:
{generated solution}
```

Tool Use. The ToolRL prompt follows GDPO's system/user layout. The system message supplies tool schemas and requires a <think> field followed by a <tool_call> block, a <response> block, or both. The user message carries the dialogue history, including tool observations in later turns.

```
System:
You are a helpful dialogue assistant capable
of leveraging tool calls to solve user tasks
and provide structured chat responses.
Available Tools: {tool list}
Steps: Think; decide on tool usage; respond.
Output Format:
<think>...</think>
<tool_call>{"name":"...","parameters":{...}}
</tool_call>
<response>...</response>

User:
Dialogue History
<user>{initial user request}</user>
<obs>{tool result, when present}</obs>
```

B.10 Protocol for the parameter mechanism analysis

The four 1.5B mathematical runs cross $\beta \in \{1/\sqrt{500}, 0.1\}$ with $\rho \in \{0.0625, 0.125\}$. Each starts from the same base model and optimizer initialization, with zero trackers and the same prompt order. Each of 12 actor updates uses 16 prompt groups with eight responses per group, a learning rate of 10^{-6}, a 2,048-token generation cap, and the 1,024-token length threshold. The projection bound is $R = 1$ and the simplex resolution is 100. For the geometric analysis, we hold fixed each of the 500 recorded Gram matrices from the 1.5B mathematical run and solve for both radii. Direction cosines are computed from those Gram matrices and the reconstructed combination coefficients. A continuous two-objective numerical solution provides the reference for the grid-resolution check. Figure 7 plots all recorded states and the analytical relative-weight kernel.

Projection remained inactive in all four runs and the recorded trajectory at $R = 1$. On the latter, the resolution-100 solver approximated continuous numerical directions with at most 0.099% relative norm error, supporting a faithful geometric interpretation of the implemented coordination.

C Extended results

C.1 Mathematical aggregation at both model scales

Table 6: Question-weighted mathematical results (%) with DeepSeek-R1-Distill-Qwen-1.5B/7B (Guo et al., 2025). The benchmarks and their sources are listed in Table 1; these aggregates weight their scores by question count. Baselines are GDPO (Liu et al., 2026a) and DVAO (Jiang et al., 2026a).

Scale	Endpoint	GDPO	DVAO	TGPO (Ours)
1.5B	Micro accuracy	37.10	36.88	**40.16**
1.5B	Micro overlength	1.12	1.62	**0.26**
7B	Micro accuracy	50.08	49.73	**50.67**
7B	Micro overlength	2.77	2.80	**1.31**

Table 6 shows that question-weighted aggregation gives TGPO the highest accuracy and lowest overlength at both scales.

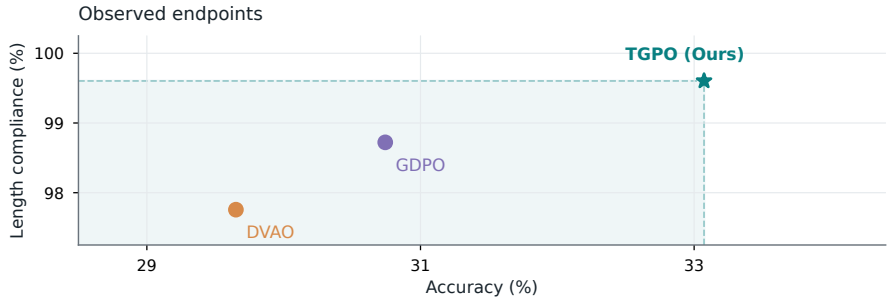

Figure 8: **Observed 1.5B mathematical endpoints.** Accuracy and length compliance are equally weighted across five suites; higher values are better on both axes. The shaded region is dominated by TGPO's point estimate.

C.2 Mechanism ablation

The BFCL-v3 ablation compares the full method with tracking or geometric coordination removed under the same 40-step protocol, as reported in Table 7. The full method reaches 35.74% overall accuracy, 87.44% strict format, and 22.88% compliant multi-turn trajectories. Without geometry, strict format is 13.64% and multi-turn format is 0%; without tracking, they are 77.21% and 10.13%. Both components contribute to structural adherence.

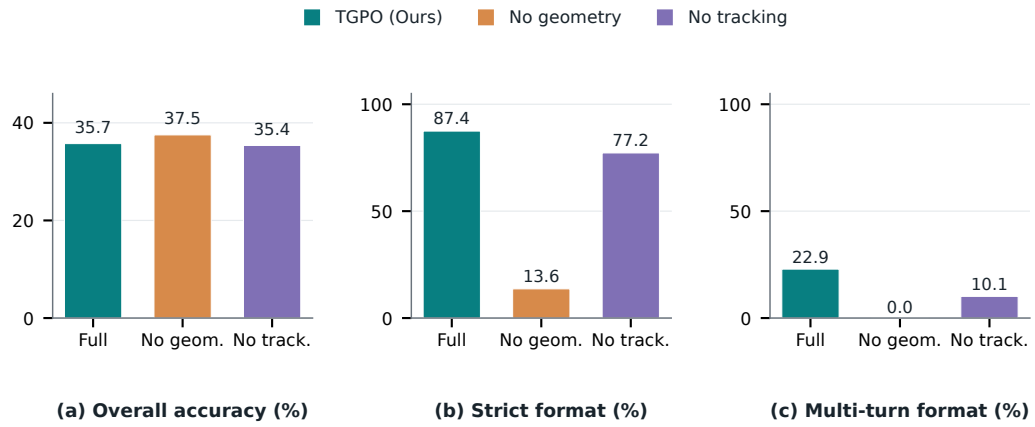

Figure 9: Mechanism ablation on the 1.5B BFCL-v3 prefix. Bars show the same 4,441-question endpoint for the full method, no-geometry, and no-tracking variants. The complete method keeps both structural endpoints high; the no-geometry point has a higher semantic accuracy estimate but loses strict trajectory compliance.

Table 7: Mechanism ablation with Qwen2.5-Instruct-1.5B (Qwen Team, 2025) on BFCL-v3 (Patil et al., 2025). All variants use 40 training steps, the same seed and decoding protocol, and all 4,441 evaluation questions. The complete method gives the strongest balanced structural result.

Variant	Overall accuracy	Strict format	Multi-turn format
Full TGPO (Ours)	35.74	**87.44**	**22.88**
No geometry	**37.52**	13.64	0.00
No tracking	35.36	77.21	10.13

D Discussion

TGPO preserves objective identity through gradient estimation, temporal tracking, and coordination. For temporal averaging $\mathcal{E}$ and coordination $\mathcal{C}$, $\mathcal{C}(\mathcal{E}(H))$ generally differs from $\mathcal{E}(\mathcal{C}(H))$. Separate states let coordination use each objective's directional history. Applying the resulting coefficients to those same states carries the estimated geometry into the update.

The evaluation brings task correctness and behavioral constraints into a common multi-reward framework. Across tasks and model scales, TGPO strengthens aggregate constraint adherence over both baselines, demonstrating the practical value of coordinating persistent objective directions. Ablations further support the complementary contributions of temporal tracking and geometric coordination.

The parameter analysis explains how these components work together. Tracking sets the memory of objective directions; the radius regulates their local coordination. Coupling through state inner products provides direct control over responsiveness and coordination strength while preserving reward definitions and the mean-direction anchor.