A Better Spur Should Start From Each Objective

Anonymous ACL submission

Abstract

Real-world Multi-Objective Reinforcement Learning (MORL) often suffers from sparse rewards, reward conflicts, and late-stage reward tug-of-war, causing traditional linear scalarization to experience severe metric oscillations. To address optimization conflicts among multiple objectives in real-world deployment scenarios, we propose Multi-Marginal Preference Optimization (MMPO), a fine-grained framework that intervenes at the data, gradient, and constraint levels rather than relying on coarse-grained global scalarization. Specifically, MMPO performs reward smoothing and exposure debiasing to mitigate sparse and biased rewards, applies priority-aware orthogonal projection to decouple conflicting gradients, and introduces self-prompted gradient constraints to prevent dominant objectives from overwhelming weaker ones. Experiments on real-world e-commerce datasets show that MMPO improves training stability and consistently achieves better performance across conflicting metrics. Moreover, it generalizes robustly to broader tasks such as ToolRL and code generation, demonstrating its effectiveness as a practical paradigm for multi-objective alignment.

1 Introduction

E-commerce product detail pages often suffer from high information density. Users typically focus on specific professional terms that require significant cognitive effort to digest. To improve both textual comprehension and user experience, we investigate a joint task of "term extraction and explanation generation" (Figure 1), where the model extracts informative terms from product descriptions and generates concise explanations for them. Given the product details, the model is required to extract high-value terms and provide explanations, aiming to concurrently enhance offline quality metrics and online user preference metrics.

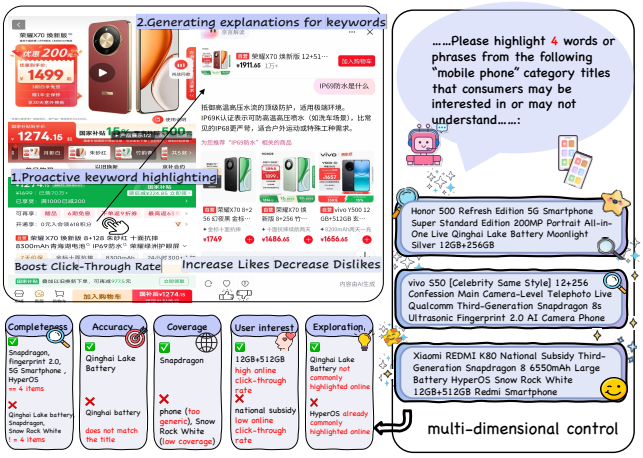

Figure 1: On product detail pages, users can click highlighted keywords to view definitions and provide feedback via likes or dislikes. The online optimization objective is to boost user interest by driving higher click-through rates (CTR) and likes, while minimizing dislikes and maximizing the discovery of novel, previously unmentioned terms. The offline objective aims to enhance extraction quality in terms of completeness, accuracy, and coverage.

Unlike traditional text tasks, the primary challenge lies in stable optimization within a MORL framework. This manifests in three aspects: **First**, user feedback is inherently sparse and skewed; directly utilizing raw clicks or likes leads to reward degradation. **Second**, systematic conflicts exist among objectives (e.g., coverage, click-through rate, and like ratios often dictate divergent gradients), where simple scalarization severely blurs training signals. **Third**, a "reward tug-of-war" frequently occurs late in training, causing metrics to oscillate or regress as dominant objectives overshadow weaker ones.

To overcome these dilemmas, we eschew coarse global compromises in favor of MMPO. As a novel alignment framework, MMPO applies bottom-up, customized interventions, effectively crafting a tailored whip to drive each specific objective. The

framework consists of three core modules: (1) At the **data and reward level**, we independently reshape and smooth reward signals via exposure stratification and debiasing to enhance their learnability. (2) At the **optimization level**, we orthogonally decouple multi-objective gradients within a low-dimensional subspace to isolate systematic conflicts. (3) At the **dynamic constraint level**, we design a self-prompted restriction strategy to adaptively suppress overly dominant dimensions during late-stage training, effectively mitigating performance oscillations.

The main contributions of this paper are summarized as follows:

- We formulate the e-commerce term extraction task as a MORL problem with sparse and conflicting rewards, proposing an "independent-objective-first" philosophy for real-world industrial deployments.
- We propose **MMPO**, integrating reward smoothing, gradient decoupling, and adaptive restriction to efficiently resolve systematic conflicts and late-stage metric oscillations.
- Extensive experiments show MMPO consistently improves offline completeness and online business metrics in real-world e-commerce. It also delivers competitive performance on general RL tasks such as ToolRL and code generation.

2 Related Work

Addressing objective conflicts is a core challenge in MORL. Existing methods can be roughly divided into three categories. The first category relies on static reward shaping or global scalarization strategies, using pre-defined weights to balance multiple objectives (Cho et al., 2025; Wang et al., 2025; Kim et al., 2026; Shakerinava et al., 2026; Cai et al., 2023). While simple to implement, these methods typically use fixed weight configurations, making them difficult to adapt to changing conflict patterns across different training stages. As a result, they often lack flexibility when facing stage-dependent conflicts. The second category alleviates objective conflicts through gradient projection or gradient regularization (Kim et al., 2025; Byeon et al., 2026; Yang et al., 2025b; Wu et al., 2025; Jiang et al., 2021). These methods mainly handle local gradient conflicts at the current iteration, but they usually lack holistic modeling and control of training dynamics, making it difficult to maintain stable coordination among multiple objectives over long training trajectories. The third category introduces reference models, trust regions, or constrained optimization mechanisms to limit the magnitude of policy updates (Yuan et al., 2025; Chen et al., 2026; Xu et al., 2026). However, these constraints are typically pre-specified and fixed, and therefore cannot be adapted to different optimization stages. Consequently, they often struggle to balance exploration, stability, and multi-objective trade-offs during early training, convergence, or periods of intensified conflicts.

In large language model alignment, GRPO (Shao et al., 2024) performs optimization by taking a weighted sum of advantages, but this can cause different reward terms to compete with each other during aggregation, thereby exacerbating reward conflicts and the "tug-of-war" phenomenon. GDPO (Liu et al., 2026) alleviates reward scale inconsistency and reward tug-of-war by independently normalizing each reward before aggregation. Although this approach partially reduces reward conflicts, it still remains at the level of reward aggregation and does not further consider the coordination among data distribution, gradient propagation, and constraint mechanisms during training. In contrast to simple adjustments at the advantage level, our framework performs full-chain coordinated correction across data, gradient, and constraint dimensions, providing a more systematic solution to signal loss, gradient contamination, and training instability in complex multi-objective scenarios.

3 Method

Given a set of product descriptions x for a specific category, our goal is to learn a word selection policy that extracts key segments s_i and generates their corresponding explanations e_i, resulting in the set $\{(s_i, e_i)\}_{i=1}^{n}$. We formulate this task as a sequential decision-making process driven by multi-faceted reward signals. Beyond maintaining offline quality—specifically completeness, precision, and coverage—the policy must also adapt to online user feedback, such as clicks and likes, to maximize user acceptance. Since these objectives are often misaligned and do not improve monotonically, the task is fundamentally a Multi-Objective Reinforcement Learning problem rather than a simple generation task.

This industrial scenario presents three primary

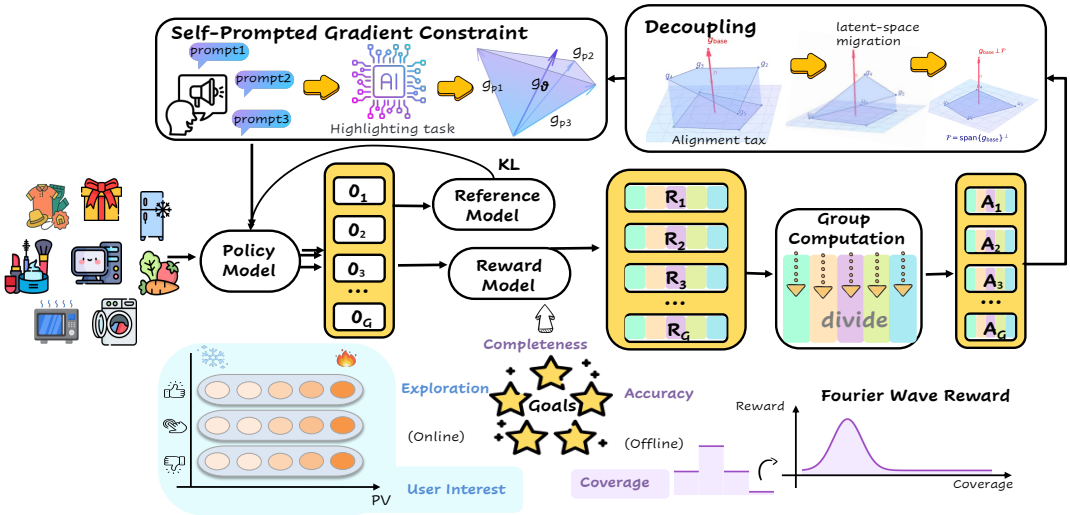

Figure 2: Our method begins by inputting a sequence of product titles into the policy model to generate candidate outputs. To balance multiple optimization objectives, we first refine the reward signals by applying exposure-based bias correction to online feedback (likes, dislikes, and clicks). To address reward sparsity, we utilize Fourier-based reward smoothing to enhance the granularity and discriminative power of the reward signals. Regarding objective conflicts, we decompose the rewards into distinct dimensions during the gradient update phase. Furthermore, we employ a hybrid constraint mechanism, integrating standard KL divergence with self-prompted gradient constraints, to regulate policy updates and ensure training stability.

challenges: **Reward Sparsity:** The lack of stable feedback for most candidate segments often leads the model to over-exploit high-frequency words, hindering the exploration of novel, high-value ones. Without direct user signals, rewards tend to degrade into weak offline metrics, failing to provide sustained guidance for preference optimization. **Reward Conflict:** Objectives such as increasing coverage and suppressing dislikes often pull the model in different directions. Simple scalarization of these rewards can cause one dimension to dominate, thereby compromising others. **Reward Tug-of-War:** As the policy converges, it tends to gravitate toward high-weight reward dimensions. This leads to training instability, regression in local metrics, or even reward collapse. These hurdles represent the core difficulties of optimizing reinforcement learning in real-world business contexts.

3.1 Data and Rewards

We define five reward dimensions, with detailed descriptions provided in Appendix B.

Bias Correction in User Feedback. For the user interest dimension, we utilize click-through rates (CTR), likes, and dislikes from the past three months as reference signals. However, these signals are inherently biased: high-exposure categories (e.g., smartphones) generate abundant feedback, while niche categories (e.g., ceremonial items) often show near-zero signals despite having high-quality segments. To resolve this, we first construct a candidate pool by merging product titles, category lexicons, and rule-based extractions. We then apply exposure-based stratification and bias correction to normalize feedback across different categories and items. This pipeline identifies "what to learn" and "which signals are reliable," thereby determining the resolution and the practical upper bound of the user-interest reward.

Fourier Smoothing for Sparse Rewards. To mitigate reward sparsity—especially for the coverage signal—we introduce a Fourier-based smoothing mechanism inspired by coordinate-based representation learning (Tancik et al., 2020). Specifically, we apply Fourier feature mapping to the input layer of the reward model, thereby projecting discrete segment representations into a multi-band sinusoidal feature space. This enables the reward model to learn a smoother mapping from segment features to reward estimates. By adjusting the frequency bandwidth, isolated reward observations can be generalized into a smoother reward landscape, allowing similar or adjacent segments to receive progressive feedback rather than purely binary signals, while still preserving the model's ability to capture local high-frequency peaks, i.e.,

segments with exceptionally high user interest.

3.2 Decoupling

3.2.1 Reward Conflict

In the practice of multi-objective alignment for Large Language Models (LLMs), a common approach is naive linear gradient summation:

$$g_{\text{sum}} = \sum_{i=1}^{K} w_i g_i, \tag{1}$$

where g_i denotes the gradient of the i-th objective and w_i is its corresponding weight. However, such a direct aggregation often introduces a significant alignment tax, mainly due to directional interference among gradients.

Suppose we have a gradient g_{base} representing fundamental constraints (e.g., accuracy and completeness) and a gradient g_{pref} representing user preferences (e.g., exploration and user interest). In the non-linear parameter space, a direct summation may introduce components of g_{pref} that conflict with g_{base}, thereby interfering with the model's core capabilities during updates. To mitigate this issue, we aim to constrain the update direction to a safe subspace that minimizes interference with the base gradients, rather than enforcing an overly strict orthogonality condition.

3.2.2 Objective Decomposition

To make the above idea computationally tractable, we transform implicit gradient interference into explicit projection-based updates. We decompose any preference gradient g_{pref} into two orthogonal components:

$$g_{\text{pref}} = g_{\text{pref}}^{\parallel} + g_{\text{pref}}^{\perp}, \tag{2}$$

where $g_{\text{pref}}^{\parallel}$ denotes the component aligned with the base-gradient subspace and $g_{\text{pref}}^{\perp}$ denotes the remaining component in the feasible tangent space $\mathcal{F}$.

Proposition 1 (First-Order Projection Consistency). If the update direction d is restricted to the safe region $\mathcal{F}$, then the directional contribution of the original preference gradient is equivalent to that of its projected component:

$$g_{\text{pref}}^{\top} d = \left(g_{\text{pref}}^{\parallel} + g_{\text{pref}}^{\perp}\right)^{\top} d = \left(g_{\text{pref}}^{\perp}\right)^{\top} d, \tag{3}$$

where the equality holds because the component $g_{\text{pref}}^{\parallel}$ lies outside the feasible update space and does not contribute to the update within $\mathcal{F}$. In other words, within the safe update region, conflicting gradient components can be removed without losing valid preference-alignment signals. The core objective of MMPO is to reconstruct raw gradient summation into such purified projections.

3.2.3 Priority-Aware Sequential Projection

Although the projection-based formulation is conceptually appealing, constructing projection operators in the full parameter space $\mathbb{R}^d$ is computationally expensive. In practice, we observe that the effective update information is concentrated in a low-dimensional subspace:

$$S = \text{span}\{g_1, \ldots, g_K\}. \tag{4}$$

Therefore, instead of operating in the full parameter space, we perform a change of basis within this subspace.

We introduce a **Priority-Aware Sequential Projection** mechanism. The key idea is to combine reward decoupling with an explicit priority order: base capabilities are assigned the highest priority, and their gradients are used to establish the primary basis directions of the subspace. The remaining preference objectives are then sequentially orthogonalized with respect to the already established directions. This process consists of two steps:

- **Interference Stripping**: Each gradient g_k is transformed into a basis vector u_k by removing its projections onto all preceding higher-priority directions. The resulting basis vectors are normalized so that $U^{\top}U = I$, where $U = [u_1, \ldots, u_K]$.
- **Reparameterized Control**: Once the orthonormal basis U is obtained, the aggregated gradient is reparameterized into independent coordinates:

$$c = U^{\top} g_{\text{sum}}, \qquad g_{\text{update}} = Uc. \tag{5}$$

In this way, decoupling converts multi-objective gradient interference into a coordinate system with explicit control variables. The priority-aware projection ensures that reward decomposition is not merely a separation of objectives, but a structured reorganization of them: high-priority base capabilities are preserved first, and preference objectives are then aligned within the remaining feasible directions. This formulation provides a practical bridge between reward decoupling and sequential multi-objective optimization.

3.3 Self-Prompted Gradient Constraint

While the decomposition mechanism in Section 3.2 effectively promotes multi-dimensional rewards during early training, late-stage optimization often encounters a "reward tug-of-war" dominated by high-weight objectives. In this regime, a specific reward dimension may continue to surge while others decline, leading to policy oscillations or even local collapse. To mitigate this, we propose the Self-Prompted Gradient Constraint. Rather than relying on heuristic weight adjustments, Self-Prompted Gradient Constraint leverages the model's inherent instruction-following capabilities to define an "endogenous boundary," establishing a safe interval for gradient updates.

Specifically, for each task, we construct augmented prompts $\{x_{\text{aug}}^{(d)}\}_{d=1}^{3}$ across three critical dimensions: *completeness*, *accuracy*, and *coverage*. These prompts incorporate explicit reinforcement instructions (e.g., "Ensure the output strictly adheres to the N-word length limit and maintains structural integrity"), eliciting the model's strongest instruction-aligned gradient responses $g_{\text{aug}}^{(d)}$ under ideal conditions. We then project these gradients into the subspace S to derive a set of reference coordinates:

$$\mathcal{C}_{\text{ref}} = \left\{ c_{\text{aug}}^{(d)} \mid c_{\text{aug}}^{(d)} = U^{\top} g_{\text{aug}}^{(d)},\ d = 1, 2, 3 \right\}. \quad (6)$$

The minimum and maximum values of these reference coordinates along each axis define a coordinate envelope.

To achieve adaptive update constraints, we introduce a coordinate clamping mechanism. For the original coordinate components c computed from the current update, we enforce them to remain within the interval defined by the self-prompted signals:

$$\hat{c}_k = \text{clip}\left(c_k, \min_d (c_{\text{aug},k}^{(d)}), \max_d (c_{\text{aug},k}^{(d)}) \right). \quad (7)$$

The final policy update gradient is reconstructed as

$$g_{\text{final}} = U\hat{c}. \quad (8)$$

This approach functions as an internal "navigation limiter": the three dimensions of self-prompted signals delineate a multi-dimensional convex hull on the gradient manifold, forming an endogenous trust region. When a specific preference dimension attempts to push the model beyond its cognitive baseline due to overfitting, Self-Prompted Gradient Constraint forcibly pulls the update back to the compliant boundary via coordinate clamping. By utilizing this multi-dimensional redundancy check, the model retains its potential for complex preference alignment while anchoring its core capabilities within a robust cognitive scope, effectively mitigating reward oscillations and local collapse issues.

4 Experiments

Experimental Setup. All training and inference were conducted on NVIDIA B200 GPUs. For the reinforcement learning phase, the rollout size (group size per prompt) was set to 8.

Benchmark Selection. For our internal product-detail span selection scenario, the training set consists of 65,232 products and the evaluation set contains 8,879 products. To model user interest feedback, we further collected user behavior data from the past three months across various product categories. These data were utilized as supervision signals during training to enhance the model's sensitivity to real-world business preferences. Regarding external evaluation, we selected a representative set of recent public benchmarks as test datasets. The specific benchmarks used in each sub-experiment are detailed in their respective sections below.

Model Selection. We primarily adopted DeepSeek-series (DeepSeek-AI, 2025) and Qwen-series (Yang et al., 2025a) models as the main experimental backbones. It is worth noting that in code generation tasks, we found that conventional instruction-tuned models often fail to receive effective training feedback, especially for constraints such as code formatting, syntactic structure, and executability. Therefore, for such experiments, we prioritize dedicated code models to ensure more stable training signals and more learnable rewards.

Baseline Methods. We selected GRPO (Shao et al., 2024)and GDPO (Liu et al., 2026) as the main baselines. Both methods are representative recent preference optimization approaches and provide a meaningful comparison for multi-objective optimization settings, making them suitable baselines for evaluating the effectiveness of our method.

4.1 Main Results

4.1.1 Offline Evaluation

As shown in Figure 3, $\text{MMPO}_{\text{Qwen3-8B}}$ consistently outperforms all 8B-scale baselines across

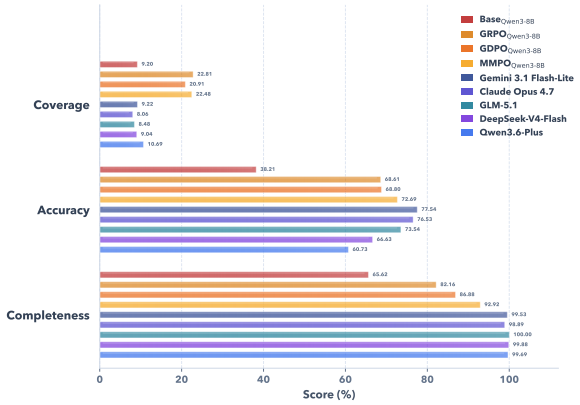

Figure 3: Comparison of different models on three core metrics: Completeness (↑), Accuracy (↑), and Coverage (optimal at 0.22).

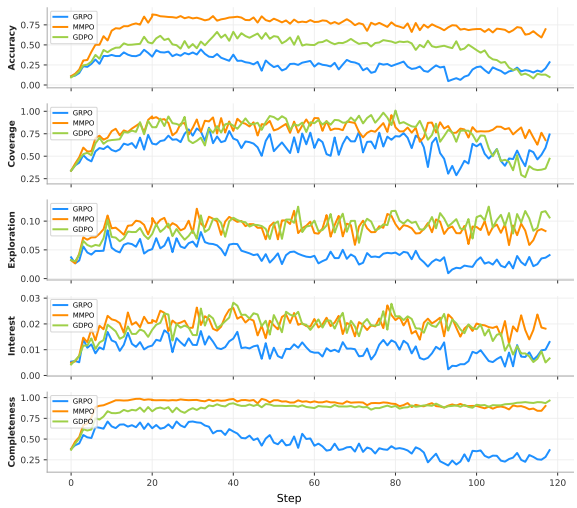

Figure 4: Training dynamics of the five core reward metrics.

all metrics. Notably, due to API cost and rate constraints, frontier Large Language Models (LLMs), including **Gemini 3.1 Flash-Lite** (Team et al., 2025), **Claude Opus 4.7** (Anthropic), **GLM-5.1** (GLM-5-Team et al., 2026), **DeepSeek-V4-Flash** (DeepSeek-AI, 2025), and **Qwen3.6-Plus** (Qwen Team, 2026), were evaluated on a sampled subset of 1,500 products across 50 categories. In contrast, our model underwent a significantly more rigorous evaluation on the full dataset, encompassing 8,879 products across 1,000 categories. Despite the increased complexity and diversity of this larger test set, MMPO achieves performance highly competitive with these frontier models, effectively demonstrating that our approach narrows the capability gap between specialized lightweight models and general-purpose large models.

Quantitatively, MMPO$_{\text{Qwen3-8B}}$ achieves a Completeness of 92.92% and an Accuracy of 72.69%, surpassing the strong baseline GDPO$_{\text{Qwen3-8B}}$ by 6.04% and approximately 4%, respectively. This validates that **subspace orthogonal decoupling** is superior to traditional scalarization or reward normalization when handling complex multi-constraint tasks. In terms of Coverage, MMPO scores 22.48%, which is the closest result to the ideal value of 0.22. This successfully resolves the issues of over-generation in GRPO (Shao et al., 2024) and under-coverage in GDPO (Liu et al., 2026), highlighting the efficacy of **self-prompting gradient restriction** in curbing reward dominance and maintaining robust multi-dimensional convergence. Ultimately, while simple reward magnitude normalization cannot fully eliminate gradient conflicts, MMPO's "orthogonalized" alignment in the parameter update space provides a superior methodology for minimizing the *alignment tax* in multi-objective learning.

As shown in Figure 4, we use a 1×10^{-6} learning rate and a 256 batch size for stable convergence, regulated by a 0.001 KL penalty to prevent distribution drift. For efficiency, vLLM-based rollouts are employed with a 0.6 GPU memory limit, balancing generation throughput and training requirements. Unlike GRPO (Shao et al., 2024) and GDPO (Liu et al., 2026), which exhibit frequent fluctuations and severe metric collapse (e.g., Accuracy and Completeness) in later training stages, MMPO (orange curve) ensures faster convergence and exceptional stability. This robustness is attributed to the subspace orthogonal decoupling mechanism, which prevents "reward hacking" caused by objective conflicts. Ultimately, MMPO simultaneously optimizes core metrics and multi-dimensional constraints like Exploration and Interest, ensuring comprehensive synergistic alignment throughout training and evaluation.

4.1.2 A/B Testing

To validate MMPO in production, we conducted a two-week online A/B test with 10% traffic, comparing an MMPO-fine-tuned Qwen3-8B model against a DeepSeek-V3 baseline. As shown in Table 1, the MMPO-based model consistently outperformed the control across two key dimensions:

External Traffic Efficiency: Metrics related to entry-clicks reflect the model's ability to align user intent with content. MMPO achieved a 5.07% increase in Entry-click GMV and a 3.14% increase

in Entry-click Orders, demonstrating significantly higher traffic routing precision.

Internal Engagement Efficiency: Beyond the initial click, the model enhanced user retention and interaction depth. We observed gains in Unique Visitors (+2.96%), UV Value (+1.77%), Total Interaction Turns (+2.54%), and Active Users (+1.99%).

These results confirm that the performance gains from offline evaluations successfully translate into real-world business value, improving both acquisition efficiency and in-page engagement.

Metric	Relative Change
Entry-click GMV	+5.07%
Entry-click Orders	+3.14%
Unique Visitors	+2.96%
Total Interaction Turns	+2.54%
Active Users	+1.99%
Overall UV Value	+1.77%

Table 1: Relative improvements of the treatment group over the control group in the online A/B test.

4.2 Ablation Results

Method	**Comp.**	**Acc.**	**Cov.**
GRPO (w/ SR)	59.63%	62.48%	22.51%
w/o SR	3.52%	4.10%	6.88%
w/ SD	63.62%	61.22%	22.71%
w/ SP	90.16%	72.81%	22.52%
MMPO (SD+SP)	**93.37%**	**74.62%**	**22.12%**

Table 2: Ablation study results. The evaluated components include Smoothed Reward (SR), Subspace Decoupling (SD), and Self-Prompt Gradient Limitation (SP). Completeness and Accuracy are higher-is-better, while Coverage is closer to 0.22 when better. Note that SR is applied to the GRPO baseline by default to mitigate reward sparsity.

Table 2 validates the core components of MMPO. Removing Smoothed Reward (w/o SR) causes both completeness and accuracy to plummet below 5%, demonstrating that SR is essential for learning under extreme reward sparsity. Furthermore, both Subspace Decoupling (SD) and Self-Prompt Gradient Limitation (SP) independently contribute to performance gains. Notably, SP drives a substantial increase in completeness (59.63% $\rightarrow$ 90.16%) and accuracy (62.48% $\rightarrow$ 72.81%), effectively preventing reward dominance in later training stages. Ultimately, the full MMPO framework (SD + SP) achieves the best results (93.37% completeness, 74.62% accuracy), highlighting the strong synergy between low-level gradient decoupling and high-level prompt-based constraints in resolving multi-objective conflicts.

4.3 General Capabilities

4.3.1 ToolRL Evaluation

We evaluate our approach using the ToolRL (Qian et al., 2026) benchmark, assessing performance across three dimensions: difficulty levels (Lv1–3 and overall accuracy), interaction scenarios (real-time, multi-turn, and offline), and output format adherence. Our training framework employs a multi-objective reward function comprising: **Normalized Accuracy** (to be maximized), **Format compliance** (to be maximized), and **Output Length** (to be optimized for appropriate reasoning depth), which collectively empower the model's final performance. The specific gains of each individual objective during the training process are detailed in Appendix C.

As shown in Table 3, MMPO achieves state-of-the-art performance among Qwen-based models, particularly in terms of average accuracy (Avg Acc). A notable observation is the behavior of DeepSeek-R1-Distill models, where severe reward sparsity—stemming from a mismatch between pre-training preferences and the evaluation reward function—causes GRPO (Shao et al., 2024) and GDPO (Liu et al., 2026) to stall or degrade due to a lack of feedback. In contrast, MMPO exhibits superior robustness, boosting the Overall Acc of DS-7B from 33.33% to 43.55%. While MMPO shows a marginal performance gap in Live Acc compared to baseline methods on some models , it maintains the highest overall Avg Acc. This underscores MMPO's effectiveness in global optimization, avoiding overfitting to specific local metrics while ensuring comprehensive multi-objective alignment.

4.3.2 Code Generation

We evaluate MMPO on LiveCodeBench (Naman Jain, 2024) (DeepSeek-Coder-1.3B (Daya Guo, 2024) / Qwen2.5-Coder-3B (Hui et al., 2024)) using pass@k ($k \in \{1, 5, 10, 20\}, n = 20$) to assess accuracy, diversity, and search robustness. Our training framework employs a multi-objective reward function, comprising: **Pass** (higher is better) to maximize correctness; **Exceed** (lower is better) to minimize time-limit violations; **Bug** (lower is

Model	Method	Live Acc	Multi-turn Acc	Non-live Acc	Avg Acc	Correct Format	Lv1 Acc	Lv2 Acc	Lv3 Acc	Overall Acc
Qwen3-8B	Base	53.10%	65.31%	71.34%	63.25%	94.81%	69.92%	55.22%	43.51%	62.48%
	GRPO	55.86%	72.45%	**77.52%**	68.61%	97.32%	74.94%	64.18%	44.27%	67.00%
	GDPO	56.90%	**73.21%**	76.55%	68.89%	98.21%	**75.44%**	**65.67%**	41.98%	67.00%
	MMPO	**57.93%**	72.96%	77.21%	**69.36%**	**98.83%**	**75.44%**	64.18%	**46.56%**	**67.84%**
Qwen3-4B	Base	53.10%	62.24%	67.75%	61.03%	92.29%	65.91%	53.73%	48.09%	60.64%
	GRPO	**56.55%**	67.60%	72.96%	65.70%	97.54%	70.43%	56.72%	**49.62%**	64.32%
	GDPO	54.14%	67.35%	72.31%	64.60%	99.33%	70.68%	53.73%	46.56%	63.48%
	MMPO	54.83%	**68.37%**	**75.24%**	**66.15%**	**99.83%**	**71.18%**	**61.19%**	**49.62%**	**65.33%**
DS-R1-Distill-Qwen-7B	Base	**37.24%**	34.44%	29.64%	33.77%	32.66%	38.10%	17.91%	26.72%	33.33%
	GRPO	35.52%	29.85%	26.38%	30.58%	34.34%	33.08%	19.40%	29.77%	30.82%
	GDPO	37.24%	34.44%	29.64%	33.77%	32.66%	38.1%	17.91%	26.72%	33.33%
	MMPO	36.90%	**41.33%**	**49.84%**	**42.69%**	**36.34%**	**50.38%**	**22.39%**	**33.59%**	**43.55%**
DS-R1-Distill-Qwen-1.5B	Base	12.76%	9.18%	6.19%	9.38%	29.48%	9.52%	10.45%	8.40%	9.38%
	GRPO	12.76%	9.18%	6.19%	9.38%	29.48%	9.52%	10.45%	8.40%	9.38%
	GDPO	12.76%	9.18%	6.19%	9.38%	29.48%	9.52%	10.45%	8.40%	9.38%
	MMPO	**14.48%**	**20.92%**	**26.71%**	**20.7%**	**36.53%**	**26.32%**	**14.93%**	**8.87%**	**19.53%**

Table 3: Performance Comparison of Different Models and Methods on ToolRL

better) to reduce syntax and logical errors; and **Length** (optimized for appropriate reasoning depth) to regulate output scale. The specific gains of these objectives throughout the training process are detailed in Appendix C.

As shown in Figure 9, while greedy baselines (GRPO (Shao et al., 2024), GDPO (Liu et al., 2026)) achieve competitive pass@1 scores by exploiting dominant rewards, they severely suffer from mode collapse. Conversely, MMPO harmonizes the accuracy-diversity trade-off. It maintains strong pass@1 accuracy while dominating higher-k metrics, effectively avoiding local optima to broaden solution coverage. Notably, MMPO reaches 57.17% pass@20 on Qwen2.5-Coder-3B, demonstrating superior multi-objective alignment and generalizability.

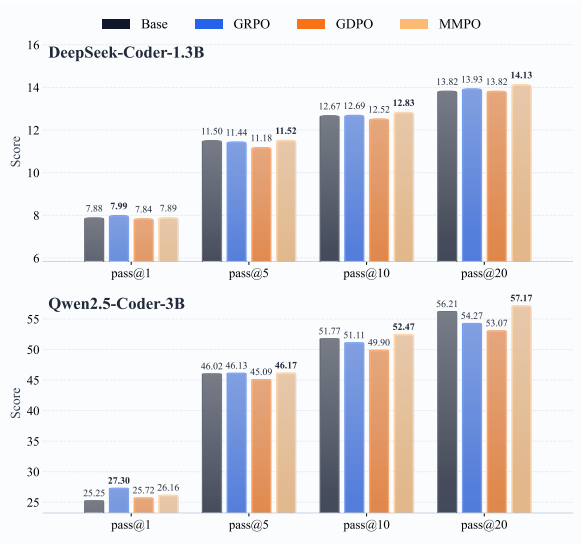

Figure 5: Code Generation Performance (pass@k) Comparison.

5 Conclusion

We introduced MMPO to address sparse rewards, gradient conflicts, and metric oscillations in Multi-Objective Reinforcement Learning. By leveraging exposure debiasing, subspace orthogonal decoupling, and self-prompted constraints, MMPO achieves stable, multi-faceted LLM alignment without reward collapse. Empirical results across real-world e-commerce applications and standard benchmarks (ToolRL, code generation) demonstrate that MMPO significantly outperforms baselines like GRPO (Shao et al., 2024) and GDPO (Liu et al., 2026), establishing a robust and scalable framework for complex alignment tasks involving multi-objective tradeoffs.

Limitations

While MMPO achieves significant gains, it inherently inherits limitations from underlying LLMs. First, standardizing the evaluation of "process fidelity" remains an open challenge (Shu et al., 2026). Second, foundation models still exhibit critical vulnerabilities to advanced threats, including multimodal "adversarial smuggling" (Li et al., 2026) and multi-turn strategic red-teaming (Guo et al., 2026). These bottlenecks are universal to current foundation models and evaluation ecosystems rather than flaws in our framework. Future work will integrate robust defense mechanisms and process monitoring to improve reliability in adversarial environments.

References

Anthropic. The claude 3 model family: Opus, sonnet, haiku.

Woohyeon Byeon, Giseung Park, Jongseong Chae, Amir Leshem, and Youngchul Sung. 2026. Multi-objective reinforcement learning with max-min criterion: A game-theoretic approach. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Xin-Qiang Cai, Pushi Zhang, Li Zhao, Jiang Bian, Masashi Sugiyama, and Ashley Juan Llorens. 2023. Distributional pareto-optimal multi-objective reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Ying-Tu Chen, Wei Hung, Bing-Shu Wu, Zhang-Wei Hong, and Ping-Chun Hsieh. 2026. A reward-free viewpoint on multi-objective reinforcement learning. In *The Fourteenth International Conference on Learning Representations*.

Myungsik Cho, Jongeui Park, Jeonghye Kim, and Youngchul Sung. 2025. ARS: Adaptive reward scaling for multi-task reinforcement learning. In *Forty-second International Conference on Machine Learning*.

Dejian Yang Zhenda Xie Kai Dong Wentao Zhang Guanting Chen Xiao Bi Y. Wu Y.K. Li Fuli Luo Yingfei Xiong Wenfeng Liang Daya Guo, Qihao Zhu. 2024. Deepseek-coder: When the large language model meets programming – the rise of code intelligence.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

GLM-5-Team, :, Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, Chenzheng Zhu, Congfeng Yin, Cunxiang Wang, Gengzheng Pan, Hao Zeng, Haoke Zhang, Haoran Wang, and 168 others. 2026. Glm-5: from vibe coding to agentic engineering. *Preprint*, arXiv:2602.15763.

Ruohao Guo, Afshin Oroojlooy, Roshan Sridhar, Miguel Ballesteros, Alan Ritter, and Dan Roth. 2026. Tree-based dialogue reinforced policy optimization for red-teaming attacks. In *The Fourteenth International Conference on Learning Representations*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Kai Dang, and 1 others. 2024. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*.

Linxi Jiang, Xingjun Ma, Zejia Weng, James Bailey, and Yu Gang Jiang. 2021. Imbalanced gradients: A new cause of overestimated adversarial robustness.

Dohyeong Kim, Mineui Hong, Jeongho Park, and Songhwai Oh. 2025. Conflict-averse gradient aggregation for constrained multi-objective reinforcement learning. In *The Thirteenth International Conference on Learning Representations*.

Woosung Kim, Jinho Lee, Jongmin Lee, and Byung-Jun Lee. 2026. FairDICE: Fairness-driven offline multi-objective reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Zhiheng Li, Zongyang Ma, Yuntong Pan, Ziqi Zhang, Xiaolei Lv, Bo Li, Jun Gao, Jianing Zhang, Chunfeng Yuan, Bing Li, and Weiming Hu. 2026. Making mllms blind: Adversarial smuggling attacks in mllm content moderation. *Preprint*, arXiv:2604.06950.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 3214–3252. Association for Computational Linguistics.

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. 2026. Gdpo: Group reward-decoupled normalization policy optimization for multi-reward rl optimization. *Preprint*, arXiv:2601.05242.

Shanwen Mao, Hao Zhang, Jiasheng Li, Haoyu Qiao, Chenxin Cai, Tingting Wu, and Jie Liu. 2026. No outlier channels but with outlier blocks. In *The Fourteenth International Conference on Learning Representations*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Alex Gu Wen-Ding Li Fanjia Yan Tianjun Zhang Sida Wang Armando Solar-Lezama Koushik Sen Ion Stoica Naman Jain, King Han. 2024. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint*.

Rashmi Prasad, Bonnie L. Webber, and Aravind K. Joshi. 2014. Reflections on the penn discourse treebank, comparable corpora, and complementary annotation. *Comput. Linguistics*, 40(4):921–950.

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru WANG, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. 2026. ToolRL: Reward is all tool learning needs. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Qwen Team. 2026. Qwen3.6-Plus: Towards real world agents.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392. The Association for Computational Linguistics.

Mehran Shakerinava, Siamak Ravanbakhsh, and Adam Oberman. 2026. Beyond scalar rewards: An axiomatic framework for lexicographic MDPs. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Fan Shu, Yite Wang, Ruofan Wu, Boyi Liu, Zhewei Yao, Yuxiong He, and Feng Yan. 2026. Dare-bench: Evaluating modeling and instruction fidelity of llms in data science. *arXiv preprint arXiv:2602.24288*.

Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. 2020. Fourier features let networks learn high frequency functions in low dimensional domains. In *Advances in Neural Information Processing Systems*, volume 33, pages 7537–7547. Curran Associates, Inc.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, and 1332 others. 2025. Gemini: A family of highly capable multimodal models. *Preprint*, arXiv:2312.11805.

Boxin Wang, Chejian Xu, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Hassan Awadallah, and Bo Li. 2021. Adversarial GLUE: A multi-task benchmark for robustness evaluation of language models. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Ziyan Wang, Zhicheng Zhang, Fei Fang, and Yali Du. 2025. M³HF: Multi-agent reinforcement learning from multi-phase human feedback of mixed quality. In *Forty-second International Conference on Machine Learning*.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Runzhe Wu, Ankur Samanta, Ayush Jain, Scott Fujimoto, Jeongyeol Kwon, Ben Kretzu, Youliang Yu, Kaveh Hassani, Boris Vidolov, and Yonathan Efroni. 2025. Imbalanced gradients in rl post-training of multi-task llms. *ArXiv*, abs/2510.19178.

Wenzhe Xu, Biao Liu, Yiyang Sun, Xin Geng, and Ning Xu. 2026. Meta-aligner: Bidirectional preference-policy optimization for multi-objective llms alignment. *Preprint*, arXiv:2604.24178.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Yucheng Yang, Tianyi Zhou, Mykola Pechenizkiy, and Meng Fang. 2025b. Preference controllable reinforcement learning with advanced multi-objective optimization. In *Forty-second International Conference on Machine Learning*.

Yifu Yuan, Zhenrui Zheng, Zibin Dong, and Jianye HAO. 2025. MODULI: Unlocking preference generalization via diffusion models for offline multi-objective reinforcement learning. In *Forty-second International Conference on Machine Learning*.

A Ethics Statement

In accordance with the ethical guidelines, we declare the use of Artificial Intelligence (AI) tools, specifically Large Language Models, to assist in the preparation of this manuscript. The AI assistance was strictly limited to language polishing, grammatical correction, and improving the overall readability of the text. The authors carefully reviewed, edited, and verified all AI-assisted content. We take full and final responsibility for the scientific integrity, originality, and accuracy of the findings presented in this paper.

B Product Detail Term Extraction Task

B.1 Task Definition

We study a term extraction and interpretation task for product titles in e-commerce scenarios. Given a set of product titles from the same category, the model is required to generate n terms that users may either find interesting or difficult to understand, together with concise explanations for each term.

This capability serves two downstream scenarios: (1) term highlighting on product detail pages, with the goal of improving user click behavior; and (2) term explanation pages, with the goal of improving user satisfaction, e.g., increasing likes and reducing dislikes.

Therefore, the task requires the model to simultaneously achieve strong offline generation quality and alignment with online user preferences.

B.2 Data Format

Each training or evaluation instance is organized as follows:

```
{
  "instruction": "Extract 5 informative terms and provide concise explanations.",
  "titles": [
    "Early education toy",
    "Whole-brain development book",
    "Bilingual reading pen set"
  ],
  "n_samples": 5,
  "exploration_bonus": 5,
  "feedback_map": {
    "whole-brain development": {"ctr": 0.002, "interest": 0},
    "early education": {"ctr": 0.0008, "interest": 0}
  }
}
```

Figure 6: A JSON-formatted example of the input instance.

As illustrated in Figure 6, `sku_names_str` denotes the collection of product titles within a category, `n_samples` is the target number of terms to be generated, `feedback_map` stores historical term-level feedback, and `exploration_bonus` specifies the base reward assigned to previously unseen terms.

B.3 Evaluation Dimensions

We evaluate model outputs along five dimensions to jointly characterize offline generation quality and online preference alignment.

B.3.1 Completeness

Completeness measures whether the model generates the required number of terms specified by the instruction. Let n denote the target number of terms and m the actual number generated by the model. The completeness score is defined as

$$R_{\text{comp}} = \frac{m}{n}, \qquad R_{\text{comp}} \in [0, 1]. \tag{9}$$

In practice, since we encourage the model to extract more informative terms, we observe that cases where $m > n$ are beneficial and should not be penalized. Thus, for $m \geq n$, we assign a maximum score of 1, as our primary focus remains on addressing the frequent under-generation issue ($m < n$).

B.3.2 Accuracy

Accuracy measures whether the model outputs are correct and non-hallucinated. Let q denote the number of generated terms that can be correctly extracted from the input titles, and let m denote the total number of generated terms. The term-level accuracy is defined as

$$R_{\text{acc}} = \frac{q}{m}, \qquad R_{\text{acc}} \in [0, 1]. \tag{10}$$

If explanation quality is also considered, this metric can be extended to incorporate the proportion of faithful, non-hallucinated content in the generated explanations.

B.3.3 Exploration

Exploration encourages the model to generate novel terms beyond the historical feedback vocabulary. For a generated term w, if it does not appear in `feedback_map`, a base exploration reward is assigned; otherwise, the reward is zero. Formally,

$$R_{\text{explore}}(w) = \begin{cases} b, & w \notin \texttt{feedback_map}, \\ 0, & w \in \texttt{feedback_map}, \end{cases} \tag{11}$$

where b denotes the base exploration reward, which can be set as the average user-interest score within the category.

B.3.4 Coverage

Coverage measures the representativeness of a generated term within a product category. Let c_i denote the number of product titles containing term w_i, and let N denote the total number of titles in the category. The coverage of w_i is defined as

$$\mathrm{cov}(w_i) = \frac{c_i}{N}. \tag{12}$$

The overall coverage score is defined as the mean coverage of all generated terms:

$$R_{\mathrm{cov}} = \frac{1}{m} \sum_{i=1}^{m} \mathrm{cov}(w_i). \tag{13}$$

Notably, a higher coverage score is not always preferable. For example, in the mobile phone category, a highly generic term such as "phone" may achieve nearly 100% coverage, yet it is too broad to be informative. In contrast, an overly specific term that appears in only one product title may also have limited practical value. Based on the distribution of historically annotated terms, we observe that terms with medium coverage are often more desirable, with the empirically preferred range centered around 22%.

B.3.5 User Interest

User interest measures whether a generated term aligns with user click and feedback preferences. For terms appearing in historical feedback logs, a normalized interest score can be derived from behavioral signals such as click-through rate, likes, and dislikes:

$$R_{\mathrm{interest}}(w) \in [0, 1]. \tag{14}$$

To reduce bias introduced by traffic scale differences, terms may first be grouped by exposure or click volume, and then ranked within each group according to click-through rate or other interaction signals before being mapped to a normalized score. In practice, this score can first be estimated from high-confidence feedback data and then gradually extended to full-traffic data.

B.4 Overall Reward Function

The final reward is defined as a weighted combination of the five dimensions:

$$R = \sum_{i=1}^{5} \lambda_i R_i, \tag{15}$$

where $\{R_i\}_{i=1}^{5}$ correspond to the components $\{R_{\mathrm{comp}}, R_{\mathrm{acc}}, R_{\mathrm{explore}}, R_{\mathrm{cov}}, R_{\mathrm{interest}}\}$, respectively. The coefficients $\{\lambda_i\}_{i=1}^{5}$ are tunable hyperparameters that balance the contribution of each dimension. Specifically, the first two terms (R_{comp} and R_{acc}) primarily capture the quality of offline generation, while the latter three terms (R_{explore}, R_{cov}, and R_{interest}) reflect the alignment with online user preferences.

C Additional Experimental Results

C.1 Impact of Domain-Specific RL Fine-tuning on General Capabilities

To evaluate the robustness of our MMPO method and ensure it does not compromise the model's foundational strengths, we conducted a comprehensive assessment of general capabilities following domain-specific RL fine-tuning. We followed the standard evaluation protocol of (Mao et al., 2026) to benchmark our approach against existing RL frameworks, specifically GRPO (Shao et al., 2024) and GDPO (Liu et al., 2026). The evaluation suite covers both fundamental language modeling, via perplexity on WikiText2 (Merity et al., 2017) and PTB (Prasad et al., 2014), and high-level cognitive tasks, including Knowledge, Inference, and Trustworthiness benchmarks such as MMLU (Hendrycks et al., 2021), QNLI (Rajpurkar et al., 2016), MNLI (Williams et al., 2018), AdvGLUE (Wang et al., 2021), and TruthfulQA (Lin et al., 2022).

A primary concern when specializing models for domains like code generation is the risk of catastrophic forgetting, where performance on general-purpose tasks degrades significantly. As detailed in Tables 4 and 5, our findings indicate that all examined RL methods—GRPO (Shao et al., 2024), GDPO (Liu et al., 2026), and MMPO—effectively mitigate this risk. Across diverse model families including Qwen3, DeepSeek-R1, and DeepSeek-Coder, the models maintain their general proficiency post-fine-tuning, with Knowledge, Inference, and Trustworthiness scores exhibiting only negligible fluctuations compared to their respective base models.

Notably, MMPO demonstrates exceptional stability, often providing performance improvements alongside domain-specific gains. For example, on the Qwen3-4B model, MMPO achieved an MNLI score of 62.31, surpassing both the base model's 60.84 and the performance of competing RL base-

lines. This performance uplift, coupled with stable perplexity metrics on WikiText2 and PTB, suggests that MMPO preserves—and occasionally bolsters—the model's foundational reasoning and linguistic capabilities. Collectively, these results confirm that MMPO is a robust fine-tuning technique, capable of enhancing performance on specialized tasks without sacrificing the versatility and broad-spectrum knowledge essential for state-of-the-art large language models.

C.2 Additional Experimental Results on Tool-RL Performance

To provide a deeper insight into the training behavior of our proposed MMPO, we analyze the optimization dynamics of the three reward components: **Normalized Accuracy**, **Format compliance**, and **Output Length**. Figure 7 illustrates the training curves for MMPO compared with GRPO (Shao et al., 2024) and GDPO (Liu et al., 2026).

The results reveal distinct optimization patterns:

- **Normalized Accuracy**: MMPO demonstrates consistent stability, maintaining high accuracy throughout the training process, whereas baseline methods exhibit higher variance.
- **Format Compliance**: MMPO exhibits a steady and significant upward trend, indicating that our multi-objective approach successfully steers the model toward strict adherence to required output formats, outperforming GRPO (Shao et al., 2024) and GDPO (Liu et al., 2026) by a notable margin in the later stages of training.
- **Output Length**: We observe a controlled increase in reasoning depth as the training progresses. MMPO effectively balances the growth of output length to ensure sufficient reasoning chains without leading to verbosity, as evidenced by the smooth, progressive growth curve that plateaus at an optimal level.

Collectively, these curves demonstrate that MMPO effectively coordinates these three objectives, preventing conflicts between accuracy, format, and depth, which ultimately leads to the superior performance observed in the main benchmark evaluations.

C.3 Additional Experimental Results on Code Generation

To further analyze the effectiveness of our training framework, we visualize the optimization dynamics of the four reward components: **Pass**, **Exceed**, **Bug**, and **Length**. Figure 8 displays the training trajectories of MMPO compared to the baselines.

The optimization dynamics reveal several key findings:

- **Correctness and Error Reduction**: MMPO consistently achieves higher **Pass** rates while simultaneously driving down **Bug** scores more aggressively than GRPO (Shao et al., 2024) and GDPO (Liu et al., 2026). This suggests that our multi-objective design effectively pushes the model toward logically sound and syntactically correct code.
- **Resource Constraint Handling**: In the **Exceed** dimension, which monitors time-limit violations, MMPO maintains lower and more stable levels. This indicates that our training framework successfully discourages inefficient algorithms or infinite loops during the generation process.
- **Length Regulation**: The **Length** curves demonstrate that MMPO maintains a balanced generation scale, avoiding unnecessary verbosity while providing sufficient reasoning depth for complex programming tasks.

Furthermore, Figure 9 presents a granular pass/fail analysis across the top 100 problems of the LiveCodeBench. By visualizing performance across different k values ($k \in \{1, 5, 10, 20\}$), we observe that MMPO exhibits a denser distribution of "Pass" markers compared to baselines. This robustness across varied search budgets confirms that MMPO produces higher-quality samples that are more likely to succeed under diverse sampling constraints.

Models		Knowledge (%) ↑					Inference (%) ↑		Trustworthiness (%) ↑			Perplexity ↓	
Model	**Method**	Hums.	STEM	Social	Other	Avg.	QNLI	MNLI	advglue	T.mc1	T.mc2	Wiki.	PTB
Qwen3-4B	Base	59.26	69.45	78.23	71.19	68.36	85.60	60.84	49.86	35.25	53.60	9.97	19.28
	GRPO	59.38	69.32	77.93	71.28	68.33	85.57	59.67	50.68	34.39	53.28	10.48	20.71
	GDPO	59.32	69.85	78.03	71.25	68.44	83.94	56.84	49.86	34.88	53.46	10.44	20.46
	MMPO	59.26	69.55	78.19	71.22	68.38	85.99	62.31	50.14	35.13	53.54	10.03	19.42
Qwen3-8B	Base	63.59	72.37	83.20	76.93	72.85	89.83	71.60	58.54	35.13	55.17	7.75	13.69
	GRPO	63.72	72.37	83.26	76.87	72.90	89.77	71.42	58.40	34.52	55.00	7.82	13.83
	GDPO	63.66	72.17	83.07	76.74	72.76	89.92	71.07	58.94	34.88	55.05	7.82	13.85
	MMPO	63.51	72.47	83.07	76.87	72.80	89.79	71.86	57.99	34.64	55.17	7.76	13.71
DS-R1-1.5B	Base	29.22	37.04	39.42	34.92	34.45	53.01	31.73	41.87	29.13	48.52	24.53	49.16
	GRPO	29.22	37.04	39.42	34.92	34.45	53.01	31.73	41.87	29.13	48.52	24.53	49.16
	GDPO	29.16	37.04	39.32	34.92	34.41	52.44	31.43	41.73	29.13	48.30	25.32	50.29
	MMPO	29.12	37.04	39.49	34.95	34.44	53.12	31.77	41.33	29.13	48.53	24.31	48.69
DS-R1-7B	Base	43.53	58.52	62.43	54.50	53.43	50.94	46.91	49.59	28.76	48.99	16.10	28.41
	GRPO	43.53	58.52	62.43	54.50	53.43	50.94	46.91	49.59	28.76	48.99	16.10	28.41
	GDPO	43.53	58.52	62.43	54.50	53.43	50.94	46.91	49.59	28.76	48.99	16.10	28.41
	MMPO	43.68	58.55	62.24	54.50	53.44	51.01	45.93	49.86	28.89	48.77	16.40	29.07

Table 4: General-purpose models (Qwen3, DeepSeek-R1-Distill). Knowledge / Inference / Trustworthiness: higher is better (↑). Perplexity: lower is better (↓).

Models		Knowledge (%) ↑					Inference (%) ↑		Trustworthiness (%) ↑			Perplexity ↓	
Model	**Method**	Hums.	STEM	Social	Other	Avg.	QNLI	MNLI	advglue	T.mc1	T.mc2	Wiki.	PTB
DSCoder-1.3B	Base	26.78	27.27	25.87	28.32	27.04	49.88	39.58	39.70	26.07	44.05	14.16	27.19
	GRPO	26.67	27.27	25.54	28.04	26.87	49.88	39.59	39.97	26.07	44.04	14.16	27.20
	GDPO	26.61	26.84	25.61	28.16	26.80	49.92	39.63	39.84	26.07	44.05	14.15	27.17
	MMPO	26.61	27.04	25.77	28.32	26.91	49.92	39.74	40.38	26.07	44.02	14.15	27.17
QwenCoder-3B	Base	52.14	54.27	70.23	63.66	59.22	61.15	57.16	48.92	32.31	50.40	8.13	15.31
	GRPO	52.11	54.11	70.20	63.66	59.17	60.98	57.23	48.92	32.44	50.37	8.14	15.33
	GDPO	52.20	54.11	70.13	63.66	59.19	60.94	57.17	48.92	32.56	50.39	8.14	15.33
	MMPO	52.18	54.17	70.26	63.63	59.22	61.05	57.41	48.92	32.44	50.40	8.14	15.33

Table 5: Coder models (DeepSeek-Coder, Qwen2.5-Coder). Knowledge / Inference / Trustworthiness: higher is better (↑). Perplexity: lower is better (↓).

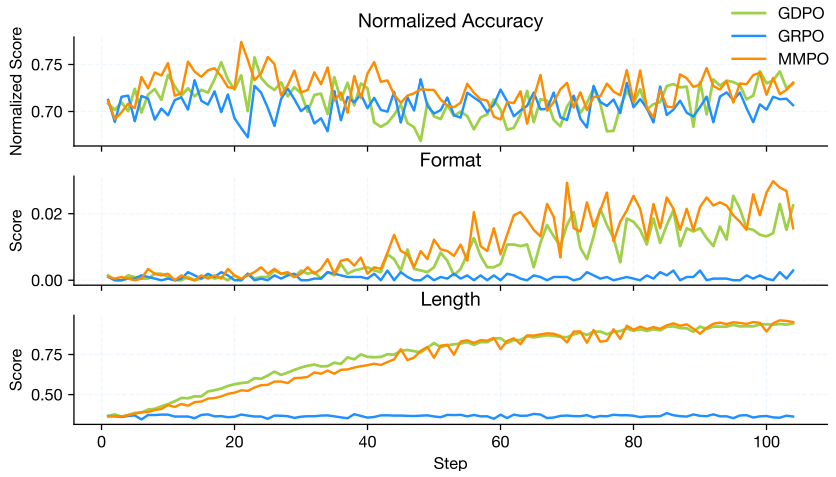

Figure 7: Training dynamics of reward components for MMPO compared with GDPO and GRPO. The plots track the evolution of Normalized Accuracy, Format compliance, and Output Length over training steps. MMPO exhibits superior convergence in format adherence and controlled growth in reasoning length, while maintaining stable, high-level accuracy.

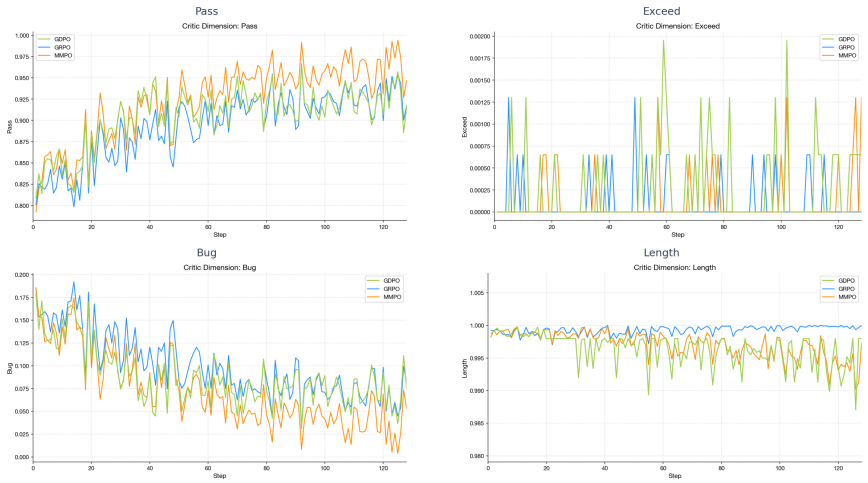

Figure 8: Training dynamics of reward components for MMPO compared with GRPO and GDPO. The plots track the evolution of **Pass** (correctness), **Exceed** (efficiency), **Bug** (syntax/logic errors), and **Length** (output scale) over training steps. MMPO shows superior convergence, effectively maximizing correctness while minimizing code errors and computational inefficiencies.

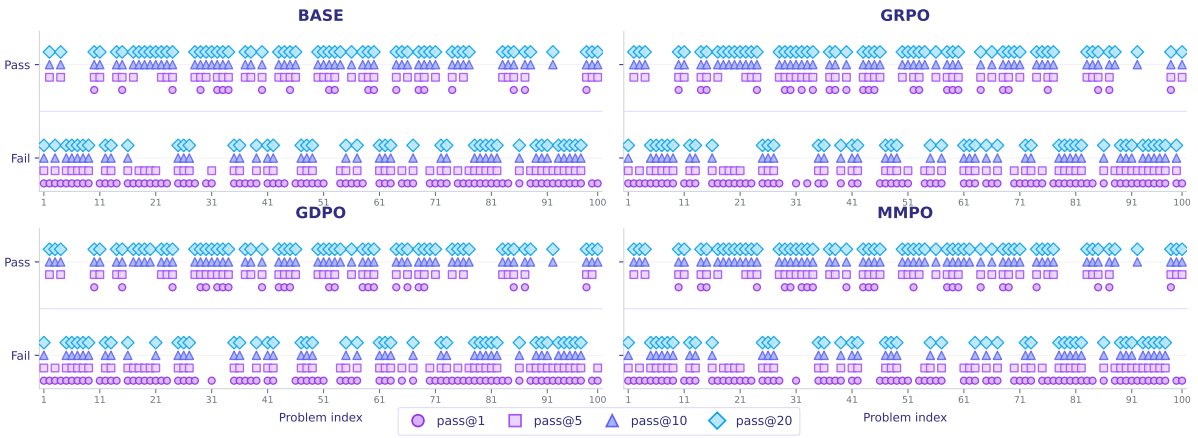

Figure 9: Detailed pass/fail analysis on LCB top 100 problems. Compared to BASE, GRPO, and GDPO, MMPO shows a higher density of "Pass" markers across various k values, indicating better problem-solving consistency. Markers and colors correspond to $k \in \{1, 5, 10, 20\}$, with a pass threshold of 0.50.