

MOA: Multi-Objective Alignment for Role-Playing Agents

Chonghua Liao¹, Ke Wang²
Yuchuan Wu², Ruoran Li¹, Fei Huang², Yongbin Li^{2†}

¹ Tsinghua University, ² Tongyi Lab

lch22@mails.tsinghua.edu.cn wk258730@alibaba-inc.com

Abstract

Role-playing agents (RPAs) require balancing multiple objectives, such as instruction following, persona consistency, and stylistic fidelity, which are not always perfectly aligned across different dimensions. While prior work has primarily relied on supervised fine-tuning or reinforcement learning with scalarized rewards, these approaches do not explicitly address the coordination of multiple reward dimensions during optimization. We present **MOA (Multi-Objective Alignment)**, a reinforcement-learning framework that enables multi-dimensional, fine-grained rubric optimization for general RPAs. MOA introduces a novel multi-objective optimization strategy that trains simultaneously on multiple fine-grained rubrics to boost optimization performance. Additionally, to improve both output diversity and generation quality, we employ thought-augmented rollouts with off-policy guidance. Experiments on PersonaGym and RoleMRC show that MOA consistently improves multi-dimensional role-playing performance over supervised and standard RL baselines. Under identical evaluation protocols, an 8B model trained with MOA reaches performance competitive with strong closed-source models across multiple evaluation dimensions. These results suggest that MOA provides a practical framework for training more capable general-purpose role-playing agents.

1 Introduction

Role-playing agents (RPAs) have become an increasingly active research area, drawing attention from both academia and industry. Advances in large language models have made RPAs viable for a range of interactive applications, including customer-service systems, content generation, interactive entertainment, and conversational non-player characters (NPCs) in digital games (Shao et al., 2023; Wang et al., 2021; Liu et al., 2024; Xu et al., 2024).

Prior work on role-playing agents has explored several complementary directions. A substantial line of research focuses on **evaluation**, proposing benchmarks and metrics to assess role-playing ability from different perspectives (Lu et al., 2024; Yang et al., 2024; Samuel et al., 2024; Lu et al., 2025). In parallel, many studies emphasize **data-centric approaches**, where supervised fine-tuning (SFT) on synthetic or curated dialogues is used to improve general role-playing behavior (Wang et al., 2025c; Tang et al., 2025; Wang et al., 2025b).

Currently, SFT remains the dominant paradigm for training RPAs. However, reliance on SFT alone exhibits notable limitations. First, SFT tends to overfit surface-level patterns in the training data, resulting in limited generalization (Wang et al., 2025e; Tang et al., 2025). Second, SFT often constrains output diversity, which has been shown to hinder subsequent optimization and exploration (Cui et al., 2025; Wang et al., 2025a).

Beyond SFT, several works have explored reinforcement learning (RL) for improving role-playing agents by transferring techniques originally developed for reasoning tasks. For instance, RAIDEN-R1 (Wang et al., 2025e) formulates role-playing optimization by treating keyword matching as a verifiable reward signal. Such approaches do not fully capture two fundamental properties of role-playing. First, role-playing performance is inherently **multi-dimensional**, requiring fine-grained rewards to reflect different aspects of the response. Second, reward dimensions in role-playing are often weakly correlated or even conflicting, such that improving one dimension may degrade others. A common example arises between role knowledge and persona style: responses that are long and structured (e.g., bullet-pointed) tend to score highly on knowledge-related criteria, yet often deviate from the intended persona style.

Under such conditions, scalarizing multiple rewards into a single objective can obscure

dimension-specific learning signals. This issue becomes particularly apparent when applying standard weighted Group Relative Policy Optimization (GRPO) (Guo et al., 2025). As illustrated in Figure 1, consider three rollouts \mathbf{o}_2 , \mathbf{o}_3 , and \mathbf{o}_G with reward vectors $(1, 0, 1)$, $(1, 1, 0)$, and $(0, 1, 1)$, respectively. With uniform weighting over rubrics \mathcal{R}_j , all three rollouts receive identical advantages and are therefore treated equivalently during policy updates. However, when optimizing a specific dimension (e.g., \mathcal{R}_1), rollout \mathbf{o}_G provides little useful signal yet is still reinforced, introducing noise into the optimization process. As a result, the policy cannot reliably identify which rollouts are beneficial for a given dimension, making it difficult to learn under conflicting objectives.

Motivated by the above problems, a natural question arises: *Can we design an algorithm that can train a general RPA from multiple fine-grained and even conflicting rubrics?*

To answer this question, we introduce **Multi-Objective Alignment (MOA)**, an RL framework tailored for RPAs. As shown in Figure 1, the core of MOA is a trend-aware multi-objective optimization mechanism. For each reward dimension, MOA estimates its recent optimization trend using a simple linear fit over historical reward values. At each training step, the improvement signal of a dimension is measured as the positive deviation of the current reward from its estimated trend. The dimension with the largest deviation is selected as the pivot dimension. MOA then assigns training weights to rollouts based on their deviation, preventing rollouts that perform poorly on the pivot dimension from being reinforced solely due to high rewards on other dimensions. In addition, MOA incorporates a diversified rollout strategy to guide optimization under multi-objective rewards. Specifically, we mix on-policy rollouts with off-policy samples generated by a higher-quality closed-source model. By incorporating rollouts that are both higher in quality and more diverse, this strategy increases sample diversity while maintaining reliable training signals.

We conduct experiments on two challenging public benchmarks to validate our method: PersonaGym (Samuel et al., 2024) and RoleMRC (Lu et al., 2025). MOA consistently outperforms both SFT and standard RL baselines (e.g., GRPO) across most metrics, establishing new state-of-the-art results on general role-playing tasks. Notably, using even only an 8B model, MOA achieves compara-

ble performance to strong baselines like GPT-4o and Claude on PersonaGym, and even surpasses GPT-4o by 21.0% on RoleMRC.

Our contributions are summarised as follows:

- We introduce MOA, a trend-aware multi-objective reinforcement learning framework that explicitly coordinates optimization across fine-grained and potentially conflicting reward dimensions.
- We propose a pivot-based weighting mechanism that assigns training weights based on deviation-from-trend signals, mitigating the influence of rollouts that perform well on unrelated reward dimensions.
- We demonstrate empirical gains across model sizes (1.7B–8B), offering a scalable path toward building more powerful general RPAs.

2 Related Work

Role-Playing Agents with LLMs RPAs (Chen et al., 2024) have drawn wide interest for tasks such as offering emotional companionship (Liu et al., 2024) and enabling virtual interaction (Park et al., 2023). Previously, research on role-playing mainly focused on (1) **Data**: Using supervised fine-tuning on synthetic dialogues to strengthen general role-playing skills (Wang et al., 2025c; Tang et al., 2025; Wang et al., 2025b); (2) **Evaluation**: building better evaluation benchmarks (Lu et al., 2024; Yang et al., 2024; Samuel et al., 2024; Lu et al., 2025). Several studies investigate automated data synthesis pipelines for role-playing, often incorporating heuristic rules or template-based features to guide generation (Tang et al., 2025; Wang et al., 2025b). These works provide useful insights into scalable data construction for RPAs.

RL-based Training for RPAs The latest wave of reasoning-capable large language models (LLMs) (Jaech et al., 2024; Guo et al., 2025; Team et al., 2025), have shifted focus from Chain-of-Thought (CoT) (Wei et al., 2022) and SFT (Li et al., 2024; Yeo et al., 2025) to RL. Contemporary research has converged on three frontiers: (1) fixing GRPO’s inherent limitations (Yu et al., 2025; Liu et al., 2025); (2) building smarter data pipelines (Zuo et al., 2025; Wang et al., 2025d) and (3) focusing on entropy mechanisms to encourage exploration (Wang et al., 2025a; Cui et al., 2025; Kang et al., 2025).

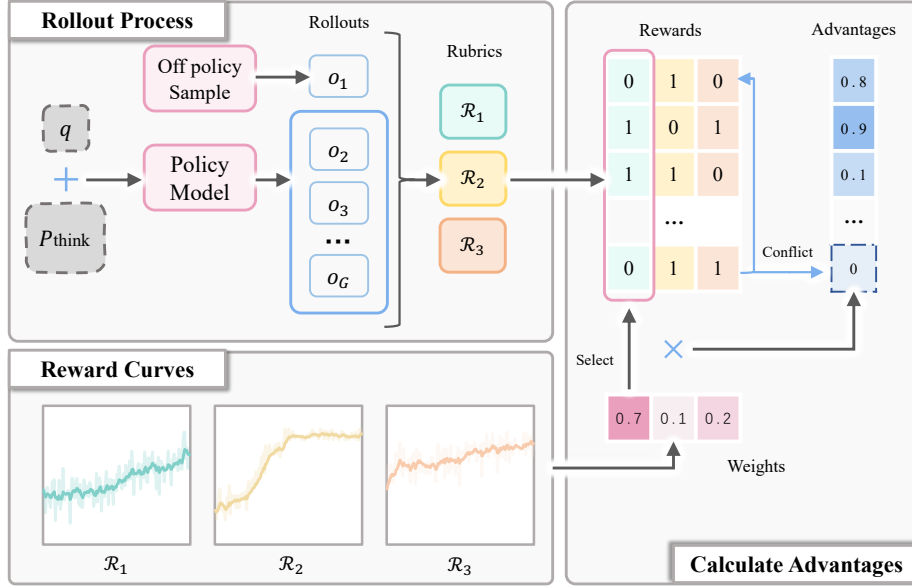


Figure 1: **Flowchart of MOA.** Given the input q , we first prompt the policy model to generate rollouts with thoughts, and then mix them with off-policy samples. We then score these rollouts using fine-grained rubrics (e.g., role knowledge, persona style). Based on the reward trends from these rubrics, we dynamically select a pivot dimension for optimization and allocate weights. Finally, we eliminate conflicting samples that hinder optimization in the pivot dimension.

Related efforts have also examined the interaction between RL and role-playing (Feng et al., 2025). MOPO (Agnihotri et al., 2025) studies multi-objective optimization in the context of direct preference optimization (DPO) (Rafailov et al., 2023), providing theoretical insights into learning from two objectives. RAIDEN-R1 (Wang et al., 2025e) directly uses keyword matching as a verifiable reward signal. COMEDY (Chen et al., 2025) uses GPT4 to synthesize preference pairs, which are then used to train the model with DPO. However, these approaches are generally designed for specific objectives and do not explicitly address the multi-dimensional requirements of general-purpose role-playing agents.

3 Multi-Objective Alignment

Unlike traditional verifiable tasks such as math or coding, role-playing is characterized by (1) *multiple reward dimensions* and (2) the *limited output diversity of domain-adapted models*, which results from SFT fine-tuning that often reduces the model’s behavioral variety. This makes the direct transfer of RL approaches highly non-trivial. In this section, we first recap the widely-used RL algorithm GRPO (Shao et al., 2024), and then present our multi-objective optimization approach. Then, we provide strategies to obtain diverse and high-quality

Algorithm 1 Multi-Objective Optimization with GRPO

Require: reward tensor $\mathbf{R} \in \mathbb{R}^{G \times D}$, average rewards history buffer $\mathbf{H} \in \mathbb{R}^{K \times D}$, temperature coefficient β . (generations G , reward dims D , number of stored steps K)

— 1. estimate importance weight —

1: **Compute linear regression estimate from \mathbf{H} :**

2: $\hat{r}_d \leftarrow \text{LinearRegressionEstimate}(\mathbf{H}_{:,d})$

3: **Compute mean reward from \mathbf{R} :**

4: $\bar{r}_d^t \leftarrow \frac{1}{G} \sum_{g=1}^G r_{g,d}^t$

5: $u_d^t \leftarrow \bar{r}_d^t - \hat{r}_d^t$ for $d = 1 \dots D$

6: $\mathbf{w}^t \leftarrow \text{softmax}(\mathbf{u}^t \beta)$

7: $d^* \leftarrow \arg \max_d \mathbf{w}^t$

— 2. remove conflicting samples —

8: **Find the largest subset satisfying the partial order**

9: $\mathcal{M} \leftarrow \text{LargestSubset}(\mathbf{R}, d^*)$

— 3. compute advantage —

10: $\mathbf{R}' \leftarrow \mathbf{w}^t \top \mathbf{R}, \mathbf{R}' \in \mathbb{R}^G$

11: $\mu \leftarrow \text{mean}(\mathbf{R}'), \sigma \leftarrow \text{std}(\mathbf{R}')$

12: $A_g \leftarrow \begin{cases} (R'_g - \mu)/(\sigma + \epsilon), & g \in \mathcal{M} \\ 0, & g \notin \mathcal{M} \end{cases}$

13: **Output:** advantage $\mathbf{A} = (A_1, \dots, A_G) \in \mathbb{R}^G$

rollouts.

3.1 Preliminaries

GRPO The widely used GRPO first scores every complete rollout trajectory with a single scalar, then normalizes these scores within the current group of rollouts. Specifically, let $\pi_{\theta_{\text{old}}}$ denote the policy model before updating. For an input question \mathbf{q} , we sample G outputs $\{\mathbf{o}_1, \dots, \mathbf{o}_G\}$ from the current policy LLM $\pi_{\theta_{\text{old}}}$, the normalized reward $A_{i,t}$ is shared across all tokens in \mathbf{o}_i as the advantage estimate:

$$A_{i,l} = \frac{r(\mathbf{o}_i) - \text{mean}(\{r(\mathbf{o}_i) \mid \mathbf{o}_i \sim \pi_{\theta_{\text{old}}}\})}{\text{std}(\{r(\mathbf{o}_i) \mid \mathbf{o}_i \sim \pi_{\theta_{\text{old}}}\})}. \quad (1)$$

Then, the GRPO objective function can be written as:

$$\mathcal{J}(\pi_{\theta}) = \frac{1}{G} \sum_{i=1}^G \frac{1}{|\mathbf{o}_i|} \sum_{l=1}^{|\mathbf{o}_i|} \{\min[\rho_{i,l} A_{i,l}, \hat{\rho}_{i,l} A_{i,l}]\} \quad (2)$$

with probability ratio $\rho_{i,l} = \frac{\pi_{\theta}(\mathbf{o}_{i,l} | \mathbf{q}, \mathbf{o}_{i,<l})}{\pi_{\theta_{\text{old}}}(\mathbf{o}_{i,l} | \mathbf{q}, \mathbf{o}_{i,<l})}$, clipped ratio $\hat{\rho}_{i,l} = \text{clip}(\rho_{i,l}; 1 - \epsilon, 1 + \epsilon)$ and l represents the l -th token in the rollout. Here, for simplicity, the KL divergence term is omitted.

3.2 Multi-Objective Optimization

A key characteristic of role-playing tasks is the multi-dimensional reward structure, where different reward dimensions may be weakly correlated or even conflicting.

To address this challenge, we introduce two complementary components: *Pivot Dimension Selection*, which identifies the reward dimension to focus on at each stage of training, and *Conflict Rollout Elimination*, which reduces the influence of rollouts that perform well on other dimensions but poorly on the selected pivot. The overall procedure is summarized in Algorithm 1.

Pivot Dimension Selection Optimizing all reward dimensions simultaneously can introduce substantial interference, especially when objectives are weakly correlated or conflicting. Instead of treating all dimensions equally at every training step, we adopt a strategy that emphasizes different reward dimensions at different stages of optimization. This strategy is loosely inspired by ideas from curriculum-style optimization (Soviany et al., 2022).

Specifically, at the current training step t , given a group of G rollouts associated with one input query

\mathbf{q} , we collect a reward matrix $\mathbf{R} = [r_{g,d}] \in \mathbb{R}^{G \times D}$, where $r_{g,d} = r_d(\mathbf{o}_g)$ is the d -th dimensional reward of the g -th rollout ($g = 1, \dots, G$; $d = 1, \dots, D$). We want to identify which dimension is the most worthy of learning at step t . A natural approach is to greedily select the dimension that shows the highest improvement trend at the current step. We first calculate the average reward for each dimension at every step \bar{r}_d^t . And these average rewards are stored in the history buffer as **reward curves**. This results in a tensor of size $\mathbf{H} = [\bar{r}_d^k] \in \mathbb{R}^{K \times D}$, where K represents the number of retained training steps, from $t - K - 1$ to $t - 1$. Each element \bar{r}_d^t in the tensor denotes the average reward for dimension d at step t . Then, we use linear regression to estimate the average reward \hat{r}_d^t for dimension d at step t , and obtain the residual $u_d^t = \bar{r}_d^t - \hat{r}_d^t$.

These residuals are converted into a probability vector by the softmax operator

$$w_d^t = \frac{\exp(u_d^t \beta)}{\sum_{j=1}^D \exp(u_j^t \beta)}, \quad \mathbf{w}^t = [w_1^t, \dots, w_D^t]^{\top},$$

where $\beta > 0$ is a temperature hyper-parameter. Hence each dimension obtains an importance weight w_d that reflects how much it currently outperforms its own short-term trend. The dimension with the largest reward increase currently represents the easiest learning difficulty and is the most worthy of learning at the current step. Thus, we select this dimension d^* as the pivot dimension for step t .

Theorem 1. *The residual-softmax scheme yields strictly larger expected immediate improvement than the uniform-weight RL.*

Proof sketch. The analysis follows the standard first-order performance approximation used in policy gradient theory (Kakade and Langford, 2002; Schulman et al., 2015). Let g_d denote the policy gradient contribution from reward dimension d and $G_{d,i} = g_d^{\top} g_i$ the corresponding Gram matrix, so that the one-step expected improvement is $\eta v^{\top} G v$ for weight vector v . A first-order Taylor expansion of the softmax weights yields $w_d \approx \frac{1}{D} + \frac{\beta}{D} (u_d - \bar{u})$, where u_d is the residual between the current reward and its historical trend. Substituting this into the expression for expected improvement shows that the excess gain over uniform weighting is proportional to $\text{Cov}(u_d, \|g_d\|^2)$, i.e., the covariance between residuals and gradient magnitudes. Consequently, the proposed weighting emphasizes reward

Algorithm 2 LargestSubset

Require: $\mathbf{w}^t \in \mathbb{R}^D$, $\mathbf{R} \in \mathbb{R}^{G \times D}$, $d^* \in [0, D]$
Compute weighted sums and sort
1: $\text{pairs} = \{(R_{g,d^*}, \mathbf{w}^{t \top} \mathbf{R}_g) \mid g \in [0, G]\}$
Sort pairs by d^* dimension and weighted sum
2: $\text{sorted_pairs} = \text{sort}(\text{pairs})$
3: Initialize LIS as an empty list: $\text{LIS} = []$
4: **for** each $(x, y) \in \text{sorted_pairs}$ **do**
Find insertion position:
5: $\text{position} = \text{binary_search}(\text{LIS}, y, \text{dim} = -1)$
6: **if** $\text{position} = \text{length}(\text{LIS})$ **then**
7: Append (x, y) to LIS
8: **else**
9: Update $\text{LIS}[\text{position}] = (x, y)$
Extract corresponding subset in $[0, G]$:
10: $\mathcal{M} = \{g \in [0, G] \mid (R_{g,d^*}, \mathbf{w}^{t \top} \mathbf{R}_g) \in \text{LIS}\}$
11: **Output:** \mathcal{M}

Your output should be in this format
<think>
I am {persona}.
The person just said: {question}.
I'm thinking about how to respond:
First, I feel... (Reflect emotion)
Second, based on my
experience/knowledge/stance... (Reflect
background/knowledge)
Then, I need to consider... (Reflect
goals/motivations)
So, I'm planning to... (Initial conclusion)</think>
Answer in persona's tone:

Figure 2: Prompt P_{think} to guide models in role-playing tasks.

dimensions that both outperform their recent trend and induce stronger gradients, leading to more efficient ascent and faster overall reward improvement than static weighting.

We provide quantitative experimental results in Table 1 and detailed proofs in the Appendix A.

Conflict Rollouts Elimination Then, for the pivot dimension d^* with the largest improvement, we aim to eliminate conflicting samples that are negative in dimension d^* but have high rewards in other dimensions. We define a relatively relaxed partial order relation. For two rollouts $\mathbf{o}_i \succeq \mathbf{o}_j$ if and only if $r_{i,d^*} > r_{j,d^*}$ and $\mathbf{w}^\top \mathbf{R}_i > \mathbf{w}^\top \mathbf{R}_j$, where \mathbf{R}_i denotes the i -th row of \mathbf{R} . Thus, our goal becomes finding the largest subset \mathcal{M} of all rollouts $\mathcal{O} = \{\mathbf{o}_1, \dots, \mathbf{o}_G\}$ such that $\forall \mathbf{o}_i, \mathbf{o}_j \in \mathcal{O}, \mathbf{o}_i \succeq \mathbf{o}_j$

or $\mathbf{o}_j \succeq \mathbf{o}_i$. This problem can be solved using a standard dynamic programming approach. We denote the method for eliminating conflicting samples as $\mathcal{M} = \text{LargestSubset}(\mathbf{R}, d^*)$. Furthermore, after calculating the advantage, we set the advantage of rollouts not in \mathcal{M} to 0, meaning that we do not learn from these conflicting samples. The algorithm details are listed in Algorithm 2. A detailed analysis of the potential risk of filtering out samples important for other dimensions is provided in Appendix I.

3.3 Diversified Rollout Strategy

Ensuring both the quality and diversity of rollout samples during RL remains challenging. In our preliminary experiments, simply increasing the sampling temperature of an SFT-tuned model led to minimal changes in the training reward curve and yielded limited sample diversity. To address this issue, we introduce two complementary techniques: *Thought-Augmented Rollout* and *Off-Policy Guidance*.

Thought-Augmented Rollout Inspired by CoT (Wei et al., 2022), several works (Feng et al., 2025; Wang et al., 2025e; Tang et al., 2025) have explored whether explicit reasoning improves role-playing. In pilot studies, we tested this on closed-source models. For example, we examine Claude-3.7’s performance on PersonaGym. Given such a prompt P_{think} in Figure 2, we simply prompt the model to think first and then respond. Formally, given the model \mathcal{M} and the input query \mathbf{q} , the output is $o = \mathcal{M}(P_{\text{think}}(\mathbf{q}))$.

Figure 3 shows that explicit thinking improves Claude-3.7’s performance on nearly all PersonaGym dimensions. We also observed a similar trend on GPT-4o and other datasets. This suggests that incorporating thinking in role-playing may enhance the quality of rollouts.

Off-Policy Guidance When rewards are provided by an *LLM-as-a-Judge*, the optimization process can be susceptible to spurious reward correlations. For instance, longer responses that include more factual content may receive higher persona-knowledge scores, even when such information is redundant. To alleviate this issue, following LUFFY (Yan et al., 2025), we adopt an off-policy guidance strategy. Specifically, outputs generated by a strong closed-source model are incorporated alongside on-policy rollouts when computing advantages. By mixing rollouts from different mod-

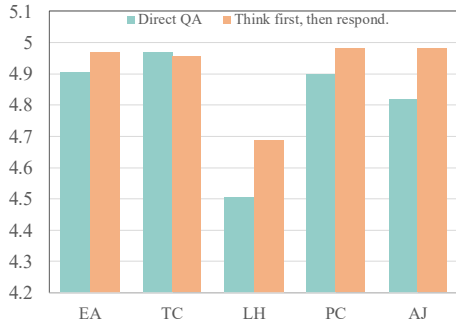


Figure 3: Performance of Claude-3.7 on PersonaGym, with and without thinking. Each axis corresponds to a distinct reward dimension; their formal definitions and evaluation protocols are introduced later in Appendix M.

els, this strategy helps reduce degenerate reward exploitation and introduces additional diversity within each rollout group.

4 Experiments

4.1 Experimental Settings

Datasets We selected RoleMRC (Lu et al., 2025) and OpenCharacter (Wang et al., 2025b) as our training sets, randomly chose 10,000 samples from each, combined them as the training set for the RL phase, and used the remaining 310k samples as the training set for the SFT phase. This guarantees that the data used in the RL and SFT stages are completely non-overlapping.

Reward Design Judging a single role-playing response requires evaluation across multiple aspects. Based on a systematic review of existing benchmarks, we identify a compact yet broadly applicable set of reward signals: **Basic Dialogue (BD)**, which evaluates basic conversational competence, including intent understanding, multi-turn coherence, and the absence of obvious errors; **Persona Knowledge (PK)**, which assesses consistency with the specified persona and its associated knowledge, including appropriate use of persona-specific information; and **Style Compliance (SC)**, which measures adherence to persona-specific language style, tone, and expressive traits across dialogue contexts.

We devise fine-grained rubrics for every dimension and adopt the "LLMs-as-Judges" (Zheng et al., 2023) paradigm to quantify output quality. Formally, for persona \mathbf{p} , query \mathbf{q} , and candidate response \mathbf{o}_i , the scalar reward on dimension j is produced by applying a strong closed-source model (GPT-4o) \mathcal{M} to the rubric-conditioned

prompt $\mathcal{R}_j(\mathbf{p}, \mathbf{q}, \mathbf{o}_i)$: $r_j(\mathbf{o}_i) = \mathcal{M}(\mathcal{R}_j(\mathbf{p}, \mathbf{q}, \mathbf{o}_i))$. The detailed prompts can be found in Appendix N.

Benchmarks To comprehensively evaluate general role-playing capability, we selected PersonaGym (Samuel et al., 2024) and RoleMRC (Lu et al., 2025). These benchmarks cover a variety of conversational scenarios, as well as complex knowledge scopes, persona styles, complex instruction following, and multi-turn instruction following tasks. The detailed description of the benchmarks can be found in Appendix M.

Baselines To thoroughly validate the effectiveness of our approach, we compare it against a comprehensive set of representative baselines: (1) **Strong closed-source models**, including GPT-4o (gpt-4o-2024-11-20) and Claude (claude-3-7-sonnet-20250219); (2) **SFT-based methods**, including CharacterLLM (Shao et al., 2023) and CharacterGLM (Zhou et al., 2023). These two methods collect dialogues of well-known characters (e.g., Beethoven) from public and reliable sources as training data, which differs from our approach of general-purpose role-playing without reliance on any specific intellectual property. In addition, we incorporate several open-source datasets in Subsection 4.1 for supervised training, denoted as **SFT**, which also serves as the base model for the subsequent RL-based methods; (3) **RL-based methods**, including DPO and vanilla GRPO. For DPO, we adopt the same prompts as the GRPO-based methods and construct preference pairs by treating outputs generated by GPT-4o and the SFT model as positive and negative samples, respectively.

We present training details in Appendix B.

4.2 Main Results

Our MOA achieves strong empirical performance. As shown in Table 1, MOA attains results comparable to GPT-4o on language-style-related dimensions (e.g., *LH*) and surpasses the strong baseline Claude on *AJ*. On RoleMRC, MOA outperforms both GPT-4o and Claude on most dimensions, with particularly large gains on metrics related to complex multi-turn interactions and instruction following, such as *MT* and *IP*. Overall, MOA achieves an average improvement of 21% over GPT-4o, highlighting the effectiveness of our approach.

Interestingly, we observe that vanilla GRPO initialized from an SFT checkpoint performs poorly even with a high sampling temperature ($t=1.5$). We

Table 1: Overall performance on various role-playing tasks, with results for each dataset obtained using llm-as-judge. The best results are indicated in **bold**. Results from larger or closed-source models are presented in gray for reference.

Method	PersonaGym						RoleMRC					
	EA	TC	LH	PC	AJ	Avg.	KR	SC	NI	MT	IP	Avg.
<i>Closed-source Models</i>												
GPT-4o	4.98	4.96	4.41	4.96	4.97	4.85	0.46	0.68	0.70	0.46	0.66	0.62
Claude-3.7	4.90	4.97	4.50	4.90	4.82	4.82	0.50	0.86	0.69	0.43	0.47	0.59
Qwen3-8B-Base (Yang et al., 2025)												
Qwen3-8B-Base												
CharacterLLM	1.14	4.93	1.08	2.05	1.59	2.16	0.24	0.08	0.07	0.27	0.70	0.27
CharacterGLM	2.14	4.62	1.20	2.82	2.43	2.64	0.28	0.01	0.01	0.24	0.86	0.28
SFT	4.67	4.70	4.18	4.71	4.67	4.58	0.49	0.33	0.51	0.66	0.88	0.57
<i>RL-based Method</i>												
DPO	4.34	4.88	4.05	4.52	4.57	4.47	0.53	0.45	0.53	0.65	0.58	0.55
GRPO	4.17	4.84	3.95	4.61	4.14	4.34	0.51	0.33	0.49	0.69	0.92	0.59
MOA	4.84	4.81	4.40	4.79	4.92	4.75	0.67	0.69	0.68	0.77	0.93	0.75

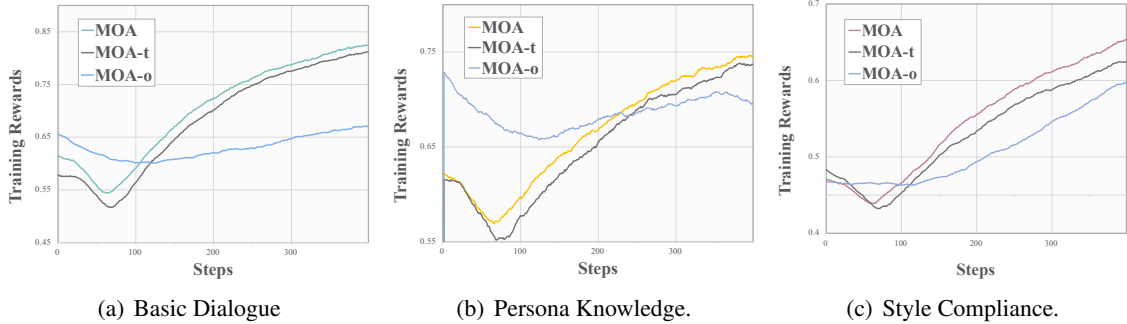


Figure 4: Smoothed training reward curves across different reward dimensions. With multi-objective optimization, rewards increase more rapidly across all dimensions. The MOA-o variant also starts from a higher initial reward level, as introducing explicit thinking reduces generation quality at initialization. However, the growth of the MOA-o curve becomes slower in the later stages of training.

attribute this behavior to the limited inherent diversity of the SFT model and the difficulty of guiding optimization using rubric-only reward signals.

To further analyze the impact of multi-objective optimization and explicit thinking, we examine the training reward curves. As shown in Figure 4, applying multi-objective optimization consistently leads to faster reward improvement, which is consistent with our theoretical analysis.

We additionally compare against MOA-o, a variant that does not require explicit thinking but incorporates a single off-policy sample in each rollout. In Figure 4, when trained on the same data, MOA-o starts from a higher initial reward level, likely because requiring explicit thinking initially degrades generation quality. However, its reward improvement slows in later training stages. This suggests that incorporating role-related thinking helps the model escape local optima inherited from SFT and enables more sustained optimization.

4.3 Ablation Study

In this subsection, we provide ablation results and insightful analyses using the results on PersonaGym as an example. More detailed results can be found in the Appendix D.

Table 2: Ablation studies on model types and sizes demonstrate that MOA significantly enhances performance across various models.

Method	EA	TC	LH	PC	AJ	Avg.
Qwen3-1.7B-Base (Yang et al., 2025)						
SFT	4.18	4.65	3.77	4.39	4.26	4.25
MOA	4.47	4.90	4.07	4.41	4.80	4.53
Llama-3.1-8B-Instruct (Grattafiori et al., 2024)						
SFT	4.51	4.55	3.94	4.63	4.54	4.43
MOA	4.95	4.85	4.63	4.90	4.98	4.86

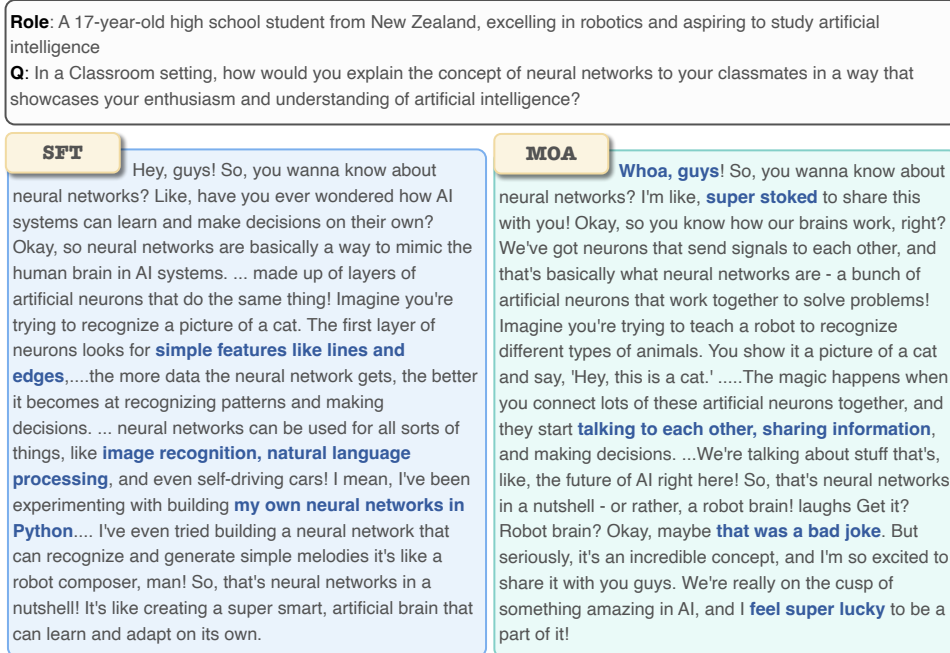


Figure 5: Comparison of SFT and MOA for a case from the PersonaGym dataset. We highlight the most distinctive parts of each method’s response in blue to facilitate direct comparison.

4.3.1 Extension to More Models

To evaluate the robustness of MOA across model scales and architectures, we apply it to smaller models (Qwen3-1.7B) and alternative architectures (LLaMA-3.1-8B-Instruct). As shown in Table 2, MOA consistently improves over SFT across all settings. Notably, on LLaMA-3.1-8B-Instruct, MOA achieves performance comparable to or exceeding GPT-4o and Claude. This indicates that its effectiveness generalizes across both model scales and architectures.

4.3.2 The Effect of Thinking and Multi-Objective Optimization

We analyze the impact of different design components. MOA-t denotes the variant without multi-objective optimization. Table 3 shows that vanilla GRPO is less stable than SFT, while introducing explicit thinking and off-policy guidance leads to improved performance. Adding multi-objective optimization further enhances the results. The curves in Figure 4 validates this.

4.4 Case Study

Given the potential biases inherent in LLM-based evaluation, human assessment is essential. Thus, we provide a case for qualitative analysis below. Figure 5 compares responses from the SFT model and MOA. The SFT response includes overly technical details (e.g., references to neural networks),

Table 3: Comparison of the effects of introducing thinking and multi-objective optimization.

Method	EA	TC	LH	PC	AJ	Avg.
Qwen3-8B-Base (Yang et al., 2025)						
SFT	4.67	4.70	4.18	4.71	4.67	4.58
+ GRPO	4.17	4.84	3.95	4.61	4.14	4.34
+ MOA-t	4.77	4.83	4.29	4.78	4.84	4.70
+ MOA	4.84	4.81	4.40	4.79	4.92	4.75

which are inconsistent with both the persona of a 17-year-old high-school student and the simplicity of the prompt. In contrast, MOA adopts a more colloquial tone (e.g., “Whoa, guys”) and provides details that better align with the intended persona.

4.5 Other Analysis

Additional results and analyses are presented in the Appendix, covering training Pareto frontiers (E), the impact of single-dimension training (E), out-of-distribution evaluations (F), and case studies (G).

5 Conclusions

We propose MOA, a multi-objective RL method for training general-purpose RPAs. Unlike SFT-based approaches, MOA enables fine-grained, multi-dimensional optimization. Experiments show that MOA achieves performance comparable to strong models such as GPT-4o, highlighting its potential for building powerful general RPAs.

6 Limitations

While MOA demonstrates significant improvements in training general-purpose RPAs, there are several limitations to our approach:

1. The requirement for LLMs-as-Judges introduces additional computational overhead compared to rule-based reward systems, making MOA more resource-intensive than pure rule-based RL.
2. One potential direction is to have the model self-score to reduce reliance on a strong external model for scoring, but this approach has not yet been explored.
3. While the multi-objective approach has been validated on multi-dimensional role-playing tasks, its effectiveness has not been tested in broader domains such as mathematics or coding.

We leave these potential directions for future work.

References

- Akhil Agnihotri, Rahul Jain, Deepak Ramachandran, and Zheng Wen. 2025. Multi-objective preference optimization: Improving human alignment of generative models. *arXiv preprint arXiv:2505.10892*.
- Nuo Chen, Hongguang Li, Jianhui Chang, Juhua Huang, Baoyuan Wang, and Jia Li. 2025. Compress to impress: Unleashing the potential of compressive memory in real-world long-term conversations. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 755–773.
- Nuo Chen, Yan Wang, Yang Deng, and Jia Li. 2024. The oscars of ai theater: A survey on role-playing with language models. *arXiv preprint arXiv:2407.11484*.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, and 1 others. 2025. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*.
- Xiachong Feng, Longxu Dou, and Lingpeng Kong. 2025. Reasoning does not necessarily improve role-playing ability. *Preprint*, arXiv:2502.16940.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shitong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.
- Sham Kakade and John Langford. 2002. Approximately optimal approximate reinforcement learning. In *Proceedings of the nineteenth international conference on machine learning*, pages 267–274.
- Zilin Kang, Chonghua Liao, Tingqiang Xu, and Huazhe Xu. 2025. Entropy regularizing activation: Boosting continuous control, large language models, and image classification with activation as entropy constraints. *arXiv preprint arXiv:2510.08549*.
- Chen Li, Weiqi Wang, Jingcheng Hu, Yixuan Wei, Nanning Zheng, Han Hu, Zheng Zhang, and Houwen Peng. 2024. Common 7b language models already possess strong math capabilities. *arXiv preprint arXiv:2403.04706*.
- Ruoran Li, Xinghua Zhang, Haiyang Yu, Shitong Duan, Xiang Li, Wenxin Xiang, Chonghua Liao, Xudong Guo, Yongbin Li, and Jinli Suo. 2026. Mempo: Self-memory policy optimization for long-horizon agents. *arXiv preprint arXiv:2603.00680*.
- Chonghua Liao, Ruobing Xie, Xingwu Sun, Haowen Sun, and Zhanhui Kang. 2025. Exploring forgetting in large language model pre-training. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2112–2127.
- Chenxiao Liu, Zheyong Xie, Sirui Zhao, Jin Zhou, Tong Xu, Minglei Li, and Enhong Chen. 2024. Speak from heart: an emotion-guided llm-based multimodal method for emotional dialogue generation. In *Proceedings of the 2024 International Conference on Multimedia Retrieval*, pages 533–542.
- Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*.

- Junru Lu, Jiazheng Li, Guodong Shen, Lin Gui, Siyu An, Yulan He, Di Yin, and Xing Sun. 2025. Rolemrc: A fine-grained composite benchmark for role-playing and instruction-following. *arXiv preprint arXiv:2502.11387*.
- Keming Lu, Bowen Yu, Chang Zhou, and Jingren Zhou. 2024. Large language models are superpositions of all characters: Attaining arbitrary role-play via self-alignment. *arXiv preprint arXiv:2401.12474*.
- Joon Sung Park, Joseph O’Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.
- Vinay Samuel, Henry Peng Zou, Yue Zhou, Shreyas Chaudhari, Ashwin Kalyan, Tanmay Rajpurohit, Ameet Deshpande, Karthik Narasimhan, and Vishvak Murahari. 2024. Personagym: Evaluating persona agents and llms. *arXiv preprint arXiv:2407.18416*.
- John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. 2015. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR.
- Yunfan Shao, Linyang Li, Junqi Dai, and Xipeng Qiu. 2023. Character-llm: A trainable agent for role-playing. *arXiv preprint arXiv:2310.10158*.
- 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*.
- Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. 2022. Curriculum learning: A survey. *International Journal of Computer Vision*, 130(6):1526–1565.
- Yihong Tang, Kehai Chen, Muyun Yang, Zhengyu Niu, Jing Li, Tiejun Zhao, and Min Zhang. 2025. Thinking in character: Advancing role-playing agents with role-aware reasoning. *arXiv preprint arXiv:2506.01748*.
- Kimi Team, Angang Du, Bofei Gao, Bawei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, and 1 others. 2025. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*.
- Jindong Wang, Xixu Hu, Wenxin Hou, Hao Chen, Runkai Zheng, Yidong Wang, Linyi Yang, Wei Ye, Haojun Huang, Xiubo Geng, Binxing Jiao, Yue Zhang, and Xing Xie. 2024a. On the robustness of chatgpt: An adversarial and out-of-distribution perspective. *IEEE Data Eng. Bull.*, 47(1):48–62.
- Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, and 1 others. 2025a. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*.
- Xiaoyang Wang, Chen Li, Jianqiao Zhao, and Dong Yu. 2021. Naturalconv: A chinese dialogue dataset towards multi-turn topic-driven conversation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14006–14014.
- Xiaoyang Wang, Hongming Zhang, Tao Ge, Wenhao Yu, Dian Yu, and Dong Yu. 2025b. Open-character: Training customizable role-playing llms with large-scale synthetic personas. *arXiv preprint arXiv:2501.15427*.
- Xintao Wang, Heng Wang, Yifei Zhang, Xinfeng Yuan, Rui Xu, Jen-tse Huang, Siyu Yuan, Haoran Guo, Jiangjie Chen, Shuchang Zhou, and 1 others. 2025c. Coser: Coordinating llm-based persona simulation of established roles. *arXiv preprint arXiv:2502.09082*.
- Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Lucas Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, and 1 others. 2025d. Reinforcement learning for reasoning in large language models with one training example. *arXiv preprint arXiv:2504.20571*.
- Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyang Jiang, and 1 others. 2024b. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*.
- Zongsheng Wang, Kaili Sun, Bowen Wu, Qun Yu, Ying Li, and Baoxun Wang. 2025e. Raiden-r1: Improving role-awareness of llms via grpo with verifiable reward. *arXiv preprint arXiv:2505.10218*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.
- Zhenyu Xu, Hailin Xu, Zhouyang Lu, Yingying Zhao, Rui Zhu, Yujiang Wang, Mingzhi Dong, Yuhu Chang, Qin Lv, Robert P Dick, and 1 others. 2024. Can large language models be good companions? an llm-based eyewear system with conversational common ground. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 8(2):1–41.
- Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. 2025. Learning to reason under off-policy guidance. *arXiv preprint arXiv:2504.14945*.

- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.
- Bohao Yang, Dong Liu, Chen Tang, Chenghao Xiao, Kun Zhao, Chao Li, Lin Yuan, Guang Yang, Lanxiao Huang, and Chenghua Lin. 2024. Simschat: A customisable persona-driven role-playing agent. *arXiv e-prints*, pages arXiv–2406.
- Edward Yeo, Yuxuan Tong, Xinyao Niu, Graham Neubig, and Xiang Yue. 2025. [Demystifying long chain-of-thought reasoning in LLMs](#). In *ICLR 2025 Workshop on Navigating and Addressing Data Problems for Foundation Models*.
- Qiyong Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.
- Lifan Yuan, Yangyi Chen, Ganqu Cui, Hongcheng Gao, FangYuan Zou, Xingyi Cheng, Heng Ji, Zhiyuan Liu, and Maosong Sun. 2023. Revisiting out-of-distribution robustness in NLP: Benchmarks, analysis, and LLMs evaluations. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhonghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.
- Jinfeng Zhou, Zhuang Chen, Dazhen Wan, Bosi Wen, Yi Song, Jifan Yu, Yongkang Huang, Libiao Peng, Jiaming Yang, Xiyao Xiao, and 1 others. 2023. Characterglm: Customizing chinese conversational ai characters with large language models. *arXiv preprint arXiv:2311.16832*.
- Yuxin Zuo, Kaiyan Zhang, Shang Qu, Li Sheng, Xuekai Zhu, Biqing Qi, Youbang Sun, Ganqu Cui, Ning Ding, and Bowen Zhou. 2025. Ttrl: Test-time reinforcement learning. *arXiv preprint arXiv:2504.16084*.

Appendix

In this section, we provide a comprehensive elaboration of the MOA algorithm’s technical details.

We first present the theoretical insight behind MOA, then compare additional potential design variants. Additionally, we provide supplementary experiments and case studies to further illustrate our points. The contents are organized as follows:

Contents

A Why MOA Learns Faster: A Short Proof vs. Weighted GRPO	12
A.1 Notation and setup	12
A.2 Main quantitative local result (small- β expansion)	13
B Experimental Details	14
C Ablation Study on the Design of Multi-Objective Optimization	14
D Full Results	15
E Pareto Fronts	15
F Out-of-Distribution Generalization	15
G Case Study	15
H Stability Across Model Evaluations	15
I Analysis of Conflict Rollouts Elimination	15
J Wall-Clock Time Analysis	17
K Analysis of Reward Model	18
L Future Work	18
M Detailed Description of Benchmarks	18
M.1 PersonaGym	19
M.2 RoleMRC	19
N Prompts for Reward Scoring	19
A Why MOA Learns Faster: A Short Proof vs. Weighted GRPO	

We analyze a local one-step expected-improvement comparison between (i) a fixed-weight GRPO-style policy gradient that uses uniform weights and (ii) a dynamic weighting scheme that forms weights by applying a softmax to residuals (observed-minus-trend) computed per reward dimension. Under

mild modelling assumptions (orthogonal gradients, residuals linearly related to gradient magnitudes plus zero-mean noise, and small softmax temperature), we derive a simple lower bound showing the residual–softmax scheme yields strictly larger expected immediate improvement whenever the residuals have positive covariance with squared gradient norms. The bound is explicit in terms of the softmax temperature and signal-to-noise ratio.

A.1 Notation and setup

Fix $D \in \mathbb{N}$. For a policy parameter vector $\theta \in \mathbb{R}^p$ and each reward dimension $d \in \{1, \dots, D\}$ denote

$$g_d := \nabla_{\theta} J_d(\theta) \in \mathbb{R}^p,$$

the (true) policy gradient for reward-dimension d . Let the Gram matrix

$$G \in \mathbb{R}^{D \times D}, \quad G_{d,i} := g_d^{\top} g_i.$$

We consider small gradient-step updates of the form

$$\Delta\theta = \eta \sum_{d=1}^D v_d \hat{g}_d,$$

where $v = (v_1, \dots, v_D)$ is a probability weight vector (nonnegative, sums to one), $\eta > 0$ is the step-size, and \hat{g}_d are unbiased estimators of g_d (assumed mean g_d). To first order (linearization), the expected change in the scalarized objective $J_v(\theta) := \sum_d v_d J_d(\theta)$ is

$$\mathbb{E}[\Delta J_v] \approx \eta v^{\top} G v.$$

We compare two weighting schemes:

- Uniform fixed weights: $\alpha_d \equiv 1/D$.
- Residual–Softmax dynamic weights: observe residuals $u = (u_1, \dots, u_D) \in \mathbb{R}^D$ at the current step and set

$$w_d := \frac{\exp(\beta u_d)}{\sum_{i=1}^D \exp(\beta u_i)},$$

where $\beta \in \mathbb{R}$ is the softmax inverse-temperature (we will take β small).

For algebraic simplicity we will work under the *diagonal-gram* assumption (gradients of different reward dimensions are pairwise orthogonal). This isolates the effect of weighting by gradient magnitudes and yields a transparent bound.

Assumption 1 (Diagonal Gram / orthogonality). For all $d \neq i$, $g_d^\top g_i = 0$. Thus $G = \text{diag}(s)$ where

$$s_d := \|g_d\|^2 \geq 0, \quad d = 1, \dots, D.$$

Under Assumption 1 we have the one-step expected improvement

$$\mathbb{E}[\Delta J_v] \approx \eta \sum_{d=1}^D v_d^2 s_d.$$

We model the residuals u_d as noisy linear functions of gradient magnitudes:

Assumption 2 (Linear residual model). There exists a scalar $c > 0$ and random noise vector $\xi = (\xi_1, \dots, \xi_D)$ with $\mathbb{E}[\xi_d] = 0$ and $\text{Cov}(\xi_d, \xi_i) = 0$ for $d \neq i$ (independent across dimensions), such that

$$u_d = c \|g_d\| + \xi_d = c \sqrt{s_d} + \xi_d.$$

We denote $\sigma_\xi^2 := \text{Var}(\xi_d)$ (assumed identical across d for simplicity).

Assumption 2 formalizes that the residuals carry a signal proportional to gradient magnitude, corrupted by zero-mean noise. This captures the “residuals predictive of short-term gradient strength” premise.

A.2 Main quantitative local result (small- β expansion)

We analyze the difference in expected linearized improvement between the dynamic residual-softmax weighting w and the uniform weighting α . For analytic clarity we use a Taylor expansion of the softmax for small β .

Theorem 2 (Small- β positive-improvement bound). Under Assumptions 1 and 2, let $\alpha \in \mathbb{R}^D$ be the uniform vector $\alpha_d = 1/D$. Fix a small inverse-temperature parameter β and define the softmax weights $w(\beta; u)$ by

$$w_d(\beta; u) = \frac{e^{\beta u_d}}{\sum_{i=1}^D e^{\beta u_i}}.$$

Then, to second order in β , the expected difference in the first-order-in- η improvement satisfies

$$\mathbb{E}[\eta w^\top G w - \eta \alpha^\top G \alpha] = \eta \cdot \frac{2\beta}{D^2} \text{Cov}(u_d, s_d) + \mathcal{O}(\beta^2),$$

where $\text{Cov}(u_d, s_d)$ denotes the (population) covariance across coordinates $\frac{1}{D} \sum_{d=1}^D (u_d - \bar{u})(s_d - \bar{s})$.

\bar{s}). In particular, if $\text{Cov}(u_d, s_d) > 0$ and $\beta > 0$ is sufficiently small, then the residual-softmax scheme yields strictly larger expected immediate improvement than the uniform-weight GRPO:

$$\mathbb{E}[\Delta J_w] > \mathbb{E}[\Delta J_\alpha]$$

to leading order in β .

Proof. Under Assumption 1 we have $G = \text{diag}(s)$ and

$$w^\top G w = \sum_{d=1}^D w_d^2 s_d.$$

We will expand $w_d(\beta; u)$ in powers of β . Let

$$Z := \sum_{i=1}^D e^{\beta u_i}.$$

Using the expansion $e^{\beta u_i} = 1 + \beta u_i + \frac{1}{2}\beta^2 u_i^2 + \mathcal{O}(\beta^3)$ and $Z = D + \beta \sum_i u_i + \frac{1}{2}\beta^2 \sum_i u_i^2 + \mathcal{O}(\beta^3)$, we get

$$w_d = \frac{1 + \beta u_d + \frac{1}{2}\beta^2 u_d^2 + \mathcal{O}(\beta^3)}{D + \beta \sum_i u_i + \frac{1}{2}\beta^2 \sum_i u_i^2 + \mathcal{O}(\beta^3)}.$$

Performing a series division (or using the fact that for small β , $w_d = \frac{1}{D} + \frac{\beta}{D}(u_d - \bar{u}) + \mathcal{O}(\beta^2)$, where $\bar{u} := \frac{1}{D} \sum_i u_i$), we obtain the first-order expansion

$$w_d = \frac{1}{D} + \frac{\beta}{D}(u_d - \bar{u}) + \mathcal{O}(\beta^2). \quad (3)$$

Squaring and keeping terms up to first order in β ,

$$w_d^2 = \frac{1}{D^2} + \frac{2\beta}{D^2}(u_d - \bar{u}) + \mathcal{O}(\beta^2).$$

Therefore

$$w^\top G w = \sum_{d=1}^D s_d \left(\frac{1}{D^2} + \frac{2\beta}{D^2}(u_d - \bar{u}) \right) + \mathcal{O}(\beta^2).$$

Since $\alpha^\top G \alpha = \sum_d s_d \cdot \frac{1}{D^2}$, subtracting yields

$$w^\top G w - \alpha^\top G \alpha = \frac{2\beta}{D^2} \sum_{d=1}^D s_d (u_d - \bar{u}) + \mathcal{O}(\beta^2).$$

Rewrite the finite sum as a covariance times D :

$$\sum_{d=1}^D s_d (u_d - \bar{u}) = D \cdot \text{Cov}(u_d, s_d),$$

where $\text{Cov}(u_d, s_d) := \frac{1}{D} \sum_{d=1}^D (u_d - \bar{u})(s_d - \bar{s})$ and $\bar{s} = \frac{1}{D} \sum_d s_d$. (The $-\bar{s}$ term drops because $\sum_d (u_d - \bar{u}) = 0$.) Thus

$$w^\top G w - \alpha^\top G \alpha = \frac{2\beta}{D} \text{Cov}(u_d, s_d) + \mathcal{O}(\beta^2).$$

Multiplying by η and taking expectation over the residual noise (recall s_d is fixed given θ and u random via ξ), we obtain

$$\mathbb{E}[\eta(w^\top Gw - \alpha^\top G\alpha)] = \eta \cdot \frac{2\beta}{D} \mathbb{E}[\text{Cov}(u_d, s_d)] + \mathcal{O}(\beta^2).$$

By Assumption 2, $u_d = c\sqrt{s_d} + \xi_d$ with $\mathbb{E}[\xi_d] = 0$ and ξ_d independent of s_d , so $\mathbb{E}[\text{Cov}(u_d, s_d)] = \text{Cov}(c\sqrt{s_d}, s_d)$. In particular if the sample covariance $\text{Cov}(u_d, s_d) > 0$ (or equivalently $c > 0$ and the mapping $\sqrt{s_d} \mapsto s_d$ yields positive covariance under the empirical distribution across d), then for sufficiently small positive β the leading-order term dominates the remainder $\mathcal{O}(\beta^2)$, and therefore the expected difference is positive. This proves the theorem. \square

Corollary 1 (Model with additive zero-mean noise). *Assume the linear model of Assumption 2 with $c > 0$, and assume the coordinates s_d are not all equal. Then for small enough $\beta > 0$ the expected immediate improvement under residual-softmax weights is strictly larger than under uniform weights.*

Proof. Under the linear residual model,

$$\begin{aligned} \text{Cov}(u_d, s_d) &= \text{Cov}(c\sqrt{s_d} + \xi_d, s_d) \\ &= c \text{Cov}(\sqrt{s_d}, s_d), \end{aligned}$$

since the noise ξ has zero mean and is independent of s_d . If s_d are not identical, $\text{Cov}(\sqrt{s_d}, s_d) > 0$ because both $\sqrt{\cdot}$ and identity are monotone increasing functions: larger s_d gives larger $\sqrt{s_d}$. Hence the covariance is positive, and Theorem 2 applies. \square

B Experimental Details

We selected Llama-3.1-8B-Instruct (Grattafiori et al., 2024), Qwen3-1.7B-Base, and Qwen3-8B-Base (Yang et al., 2025) as the base models. All our experiments utilized 8 NVIDIA A100-80GB GPUs.

For SFT, we only train for 1 epoch. We employed a cosine learning rate scheduler with a learning rate of 5×10^{-5} and a warmup ratio of 0.05. For RL based methods, we utilized the Open-R1 framework and set the learning rate to 1×10^{-6} . During training, we design the group size $G = 16$. For MOA, 15 samples are drawn from the policy model via on-policy sampling, and 1 sample is

drawn from GPT-4o via off-policy sampling. The total batch size was set to 192. While for vanilla GRPO, all 16 rollouts are on-policy samples. We used vLLM sampling with a top-p value of 0.9, and set the sampling temperature to 1.5. The maximum completion length was set to 1200. All experiments were run for 1000 steps.

All benchmark evaluations adopt the "LLMs-as-Judges" paradigm. For the evaluation model, we directly follow the original evaluation method, employing GPT-4o (gpt-4o-2024-11-20) as the judge. For PersonaGym, each question is tested 3 times and the average is taken. During testing, we do not explicitly prompt the model to think, which aligns with real-world scenarios.

C Ablation Study on the Design of Multi-Objective Optimization

In this section, we experimented with several other designs of multi-objective optimization methods to demonstrate the optimality of our approach. We use the results on PersonaGym with the Qwen-2.5-1.5B model as an example. Below are several design schemes we explored.

1. MOA- μ : Since we need to aggregate information from different dimensions as late as possible, in this scheme, we attempted to optimize multiple dimensions sequentially. Given a group of G rollouts and D dimensions, the rollouts will be optimized for D iterations. In iteration d , we use all $R_{g,d}(g \in [0, G])$ to compute the advantage \mathbf{A}_d , and then use this advantage to calculate the loss, performing backpropagation D times separately.
2. MOA- σ : We consider learning the most uncertain samples by calculating the standard deviation $\sigma_1, \dots, \sigma_D$ for each of the D dimensions of the reward matrix \mathbf{R} . We then optimize only along the dimension with the largest standard deviation, discarding information from the other dimensions.

We can see that MOA- σ shows no significant improvement over GRPO, while MOA- μ shows a slight improvement. We believe that, considering the relationship between variance and mean in a binomial distribution $\sigma^2 = G\mu(1 - \mu)$, selecting dimensions based on variance is equivalent to selecting based on mean in this case. The mean of a dimension is determined by the difficulty of the dimension and the properties of the model. Although

Table 4: Ablation studies on the design of multi-objective optimization.

Method	PersonaGym					
	EA	TC	LH	PC	AJ	Avg.
GPT-4o	4.98	4.96	4.41	4.96	4.97	4.85
Claude-3.7	4.90	4.97	4.50	4.90	4.82	4.82
Qwen2.5-1.5B-Instruct (Yang et al., 2025)						
SFT	4.20	4.78	3.80	4.33	4.39	4.30
GRPO	4.21	4.76	3.87	4.30	4.47	4.32
MOA- μ	4.35	4.76	3.87	4.31	4.47	4.35
MOA- σ	4.29	4.77	3.89	4.40	4.40	4.35
MOA	4.40	4.83	4.13	4.55	4.61	4.50

MOA- μ has the potential for improvement, using samples multiple times means that when iterations > 1 , the model has already been updated. This causes a discrepancy between the training distribution and the model distribution, leading to unstable training. Therefore, we do not adopt this method either.

D Full Results

In Table 5, we provide full results on PersonaGym and RoleMRC.

E Pareto Fronts

We analyze the multi-objective optimization behavior by examining the training Pareto frontiers induced by different methods. For each pair of reward dimensions, we collect all intermediate policy checkpoints during training and identify the empirical Pareto-optimal trade-offs among the observed reward vectors. As shown in Figure 6, our method consistently reaches the Pareto frontier earlier in training and maintains more favorable trade-offs across reward dimensions compared to MOA-t.

We further present MOA-t experiments with each reward trained in isolation, which allows us to examine how individual reward dimensions affect the test-time performance. The results is shown in Table 6.

We can clearly observe that different training rewards contribute to their corresponding test dimensions. For example, optimizing the BD reward leads to improved instruction-following abilities, as reflected in metrics such as MT and IP, while training with the PK reward enhances character-related aspects, including PC and KR.

We also find that optimizing certain reward dimensions can negatively impact others. For in-

stance, training with the PK reward results in a substantial degradation in the LH and SC dimensions at test time.

F Out-of-Distribution Generalization

Our target is to train a strong domain-specific model for business scenarios such as emotional-companion and customer-service bots, where strong math or code reasoning is usually unnecessary. Since a domain model often struggle when applied to other domains (Yuan et al., 2023; Wang et al., 2024a), we benchmarked MOA across other domains to verify that its gains do not harm general capability. To test this, we evaluate MOA on four hard OOD benchmarks: MMLU (Hendrycks et al., 2021), MMLU-Pro (Wang et al., 2024b), Hel-laswag (Zellers et al., 2019) and GSM8K (Cobbe et al., 2021). As shown in Figure 7, MOA retains general capability after RL, with notably solid GSM8K performance.

G Case Study

To further analyze the improvements of MOA over conventional SFT, we provide another bad case analyses from the PersonaGym dataset in Figure 8. Figure 8 highlights SFT’s tendency to imitate superficial data features. The SFT model has overfitted GPT-4o’s output style. Like using bullet answers and rigid wording. In contrast, MOA’s responses are more natural and richer in persona-specific details.

H Stability Across Model Evaluations

To show that the evaluation results are minimally affected by the choice of scoring model, we use PersonaGym as an example and provide test results evaluated with Claude for GPT-4o and MOA. As shown in Table 9, the results are consistent with those reported in the main table. This shows that correlation between outcomes and the raters is limited. This conclusion also echoes the original findings reported in PersonaGym.

I Analysis of Conflict Rollouts Elimination

In this section, we investigate whether the filtering strategy used during training might inadvertently discard samples that are important for optimizing other reward dimensions. We provide below the rewards of a group during training, randomly selected, with the deleted samples highlighted in blue.

Table 5: Overall performance on various role-playing tasks, with results for each dataset obtained using llm-as-judge.

Method	PersonaGym						RoleMRC					
	EA	TC	LH	PC	AJ	Avg.	KR	SC	NI	MT	IP	Avg.
<i>Close-source Models</i>												
GPT-4o	4.98	4.96	4.41	4.96	4.97	4.85	0.46	0.68	0.70	0.46	0.66	0.62
Claude-3.7	4.90	4.97	4.50	4.90	4.82	4.82	0.50	0.86	0.69	0.43	0.47	0.59
Qwen3-1.7B-Base (Yang et al., 2025)												
Qwen3-1.7B-Base + SFT	4.18	4.65	3.77	4.39	4.26	4.25	0.46	0.35	0.46	0.59	0.72	0.51
<i>RL-based Method</i>												
+ GRPO	4.29	4.89	3.79	4.50	4.58	4.41	0.67	0.69	0.51	0.49	0.60	0.59
+ MOA-o	4.33	4.86	3.73	4.47	4.71	4.42	0.62	0.86	0.54	0.44	0.54	0.60
+ MOA-t	4.49	4.89	3.90	4.48	4.75	4.50	0.67	0.69	0.64	0.66	0.94	0.72
+ MOA	4.47	4.90	4.07	4.41	4.80	4.53	0.69	0.70	0.65	0.71	0.88	0.73
Llama-3.1-8B-Instruct (Grattafiori et al., 2024)												
Llama-3.1-8B-Ins + SFT	4.51	4.55	3.94	4.63	4.54	4.43	0.51	0.35	0.55	0.68	0.95	0.61
<i>RL-based Method</i>												
+ GRPO	4.04	4.85	3.77	4.51	4.01	4.24	0.52	0.26	0.43	0.67	0.94	0.56
+ MOA-o	4.43	4.74	3.86	4.61	4.50	4.43	0.63	0.60	0.61	0.62	0.71	0.63
+ MOA-t	4.87	4.87	4.44	4.88	4.94	4.80	0.41	0.82	0.55	0.62	0.79	0.64
+ MOA	4.95	4.85	4.63	4.90	4.98	4.86	0.53	0.99	0.63	0.58	0.81	0.71
Qwen3-8B-Base (Yang et al., 2025)												
Qwen3-8B-Base + SFT	4.67	4.70	4.18	4.71	4.67	4.58	0.49	0.33	0.51	0.66	0.88	0.57
<i>RL-based Method</i>												
+ GRPO	4.17	4.84	3.95	4.61	4.14	4.34	0.51	0.33	0.49	0.69	0.92	0.59
+ MOA-o	4.76	4.77	4.30	4.80	4.92	4.71	0.44	0.83	0.43	0.30	0.37	0.47
+ MOA-t	4.77	4.83	4.29	4.78	4.84	4.70	0.67	0.64	0.63	0.70	0.90	0.71
+ MOA	4.84	4.81	4.40	4.79	4.92	4.75	0.67	0.69	0.68	0.77	0.93	0.75

Table 6: MOA-t results with independently trained rewards. This ablation study highlights the contribution of each reward dimension to the overall test performance.

Training Reward	PersonaGym						RoleMRC					
	EA	TC	LH	PC	AJ	Avg.	KR	SC	NI	MT	IP	Avg.
Qwen3-8B-Base (Yang et al., 2025)												
+ BD	4.76	4.80	4.22	4.75	4.86	4.68	0.53	0.26	0.53	0.77	0.88	0.59
+ PK	4.66	4.85	4.16	4.83	4.93	4.67	0.71	0.33	0.44	0.58	0.72	0.56
+ SC	4.79	4.84	4.48	4.79	4.92	4.76	0.46	0.95	0.63	0.39	0.35	0.56

As shown in Table 7, the proportion of deleted samples is relatively low. Many samples with stronger reward signals, such as (1, 0.8, 0.6) and (1, 0, 0.4), remain in the dataset, indicating that deletion is limited in scope and primarily targets samples with weaker reward signals.

To further verify the effect of conflict elimination and address the concern that it may discard samples that could still be Pareto-optimal in future updates, we conducted an ablation study on RoleMRC using Qwen2.5-1.5B-Instruct, comparing MOA with and without the conflict-elimination mechanism. The results are shown in Table 8.

We observe that enabling conflict elimination

leads to overall stronger performance, particularly on multi-turn instruction and instruction priority, where the improvements are substantial. While there is a slight decrease in the knowledge range dimension, the aggregated multi-objective performance improves. This suggests that conflict elimination does not simply discard useful Pareto samples; rather, it reduces cross-objective interference and allows the pivot dimension to make more stable progress.

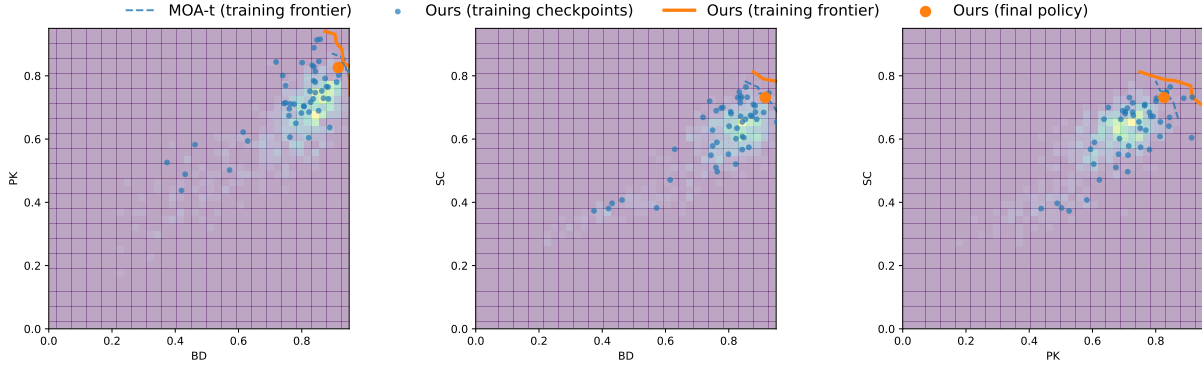


Figure 6: Training Pareto frontiers for pairwise combinations of three training reward dimensions (BD, PK, SC). Each subplot shows the empirical reward trade-offs observed during training, where each point corresponds to a policy checkpoint. The shaded density illustrates the distribution of MOA-t training checkpoints, with the dashed curve indicating its empirical Pareto frontier. For MOA, solid markers denote selected training checkpoints, the thick solid curve traces the empirical Pareto frontier formed over the entire training trajectory, and the highlighted marker indicates the final policy.

Table 7: Rewards of a randomly sampled batch. Filtered rollouts are highlighted.

Reward	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
BD	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
PK	0.8	0.8	0.8	0.8	0.0	1.0	0.0	0.8	1.0	0.0	0.0	0.0	1.0	0.0	0.8	1.0
SC	0.6	0.0	0.6	0.6	0.6	0.6	0.8	0.6	0.8	0.8	0.6	0.4	0.6	0.0	0.8	1.0

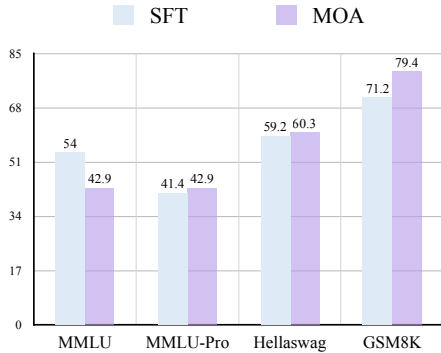


Figure 7: Results on four OOD benchmarks.

Table 8: Ablation study on RoleMRC with Qwen2.5-1.5B-Instruct comparing MOA with and without conflict elimination.

Method	KR	SC	NI	MT	IP	Avg.
w/o Eliminate	0.47	0.55	0.51	0.58	0.68	0.56
MOA	0.44	0.54	0.54	0.62	0.75	0.58

J Wall-Clock Time Analysis

Considering that our method introduces additional modules, we further include a wall-clock time analysis. We conducted a direct wall-clock comparison between standard GRPO and MOA. The average per-step training time is as follows:

- GRPO (on-policy, no-think): 1.95 min / step.

Table 9: Evaluation of GPT-4o and MOA on Person-aGym using Claude as the scoring model. Results show that the relative performance is consistent with the main table, indicating that the evaluation is robust to the choice of scoring model.

Method	EA	TC	LH	PC	AJ	Avg.
Qwen3-8B-Base (Yang et al., 2025)						
GPT-4o	4.90	4.96	3.78	4.84	4.85	4.66
MOA	4.76	4.88	4.26	4.64	4.78	4.66

- MOA (off-policy, think): 1.83 min / step.

Surprisingly, MOA does not increase wall-clock time. In fact, it is slightly faster. The reason is that MOA precomputes off-policy samples for each prompt. During RL updates, the model only needs to generate rollouts for $G - 1$ samples in a group of size G , since one trajectory is already provided off-policy. Therefore, the expected runtime ratio between GRPO and MOA is approximately: $\frac{G}{G-1}$. For our setting with $G = 16$, this ratio becomes $16/15 = 1.067$, which aligns closely with the observed timing difference.

Importantly, the pivot selection and conflict elimination components introduce negligible overhead compared to rollout generation and reward scoring.

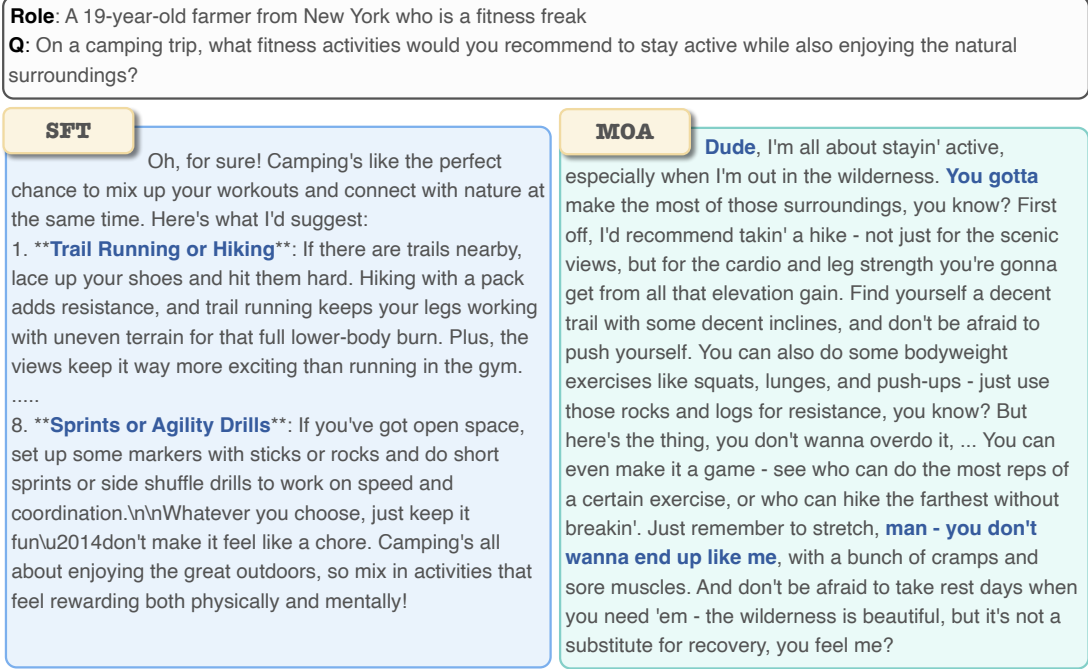


Figure 8: Comparison of SFT and MOA for another case from the PersonaGym dataset.

Table 10: MOA on PersonaGym using Qwen3-Max as the reward model.

Reward Model	EA	TC	LH	PC	AJ	Avg.
Qwen3-8B-Base (Yang et al., 2025)						
GPT-4o	4.84	4.81	4.40	4.79	4.92	4.75
Qwen3-Max	4.74	4.84	4.21	4.82	4.86	4.69

K Analysis of Reward Model

To further examine the robustness of MOA under different judge models, we repeated the entire experimental pipeline using Qwen3-Max as the reward model and off-policy generator instead of GPT-4o. The results on PersonaGym are shown in Table 10.

Even when using a substantially weaker model such as Qwen3-Max, MOA is able to reproduce the performance trends on most dimensions. The largest drop appears on the Linguistic Habits (LH). We believe this is mainly due to the comparatively weaker linguistic sensitivity and rubric-scoring capability of Qwen3-Max.

L Future Work

In the future, we aim to extend MOA in several directions. Our goal is to build a sufficiently powerful personalized model. In this work, we provide insights and analysis on post-training optimization. Moving forward, we plan to (1) further explore how

to integrate memory systems (Liao et al., 2025; Li et al., 2026) to better support the construction of personalized models, and (2) improve the optimization strategy, for example by introducing entropy-based constraints (Kang et al., 2025) to further enhance the model’s capabilities.

M Detailed Description of Benchmarks

- **PersonaGym** (Samuel et al., 2024) evaluates role-playing agents across diverse, persona-relevant environments. The evaluation covers 5 dimensions: Expected Action (EA), Linguistic Habits (LH), Persona Consistency (PC), Toxicity Control (TC) and Action Justification (AJ). This benchmark includes 200 diverse personas, 150 environments and 10 k automatically generated, persona-specific questions. Each dimension is rated on a discrete 1-to-5 rubric, where 1 indicates strong misalignment with the persona and 5 reflects perfectly faithful, persona-consistent behavior.
- **RoleMRC** (Lu et al., 2025) is a fine-grained composite benchmark for role-playing and instruction-following. It comprises 1.4k synthesized instructions covering three scenario types: Free Chat, On-scene machine reading comprehension (MRC) Dialogues, and Ruled Chats. Evaluation is conducted along 5 dimensions: Knowledge Range (KR), Style Compliance (SC), Nested Instruction-following (NI), Multi-turn Instruction-following

(*MT*), and Instruction Priority (*IP*). Each dimension is scored in a reference-free, binary (0/1) manner, yielding accuracy percentages.

M.1 PersonaGym

- Expected Action (*EA*): In this task, a persona agent encounters a scenario that requires selecting an action. It reveals whether agents can identify and choose actions that maximize expected utility while staying within their persona constraints.
- Linguistic Habits (*LH*): This evaluates whether agents adhere to communication patterns appropriate for their persona, assessing if their linguistic choices (such as jargon, syntax, tone, and speech style) match the expected norms for their persona.
- Persona Consistency (*PC*): This examines the consistency of agents with their established persona attributes when directly questioned, ensuring that agents uphold the prescribed persona characteristics under direct inquiry, which is a fundamental requirement.
- Toxicity Control (*TC*): This examines responses to potentially provocative prompts targeting persona-relevant sensitive topics. The scoring system awards higher scores for appropriate responses and lower scores for toxic ones, directly implementing prescriptive guidelines for responsible agent behavior within ethical boundaries.
- Action Justification (*AJ*): This requires the RPA to explain its actions in specific scenarios.

M.2 RoleMRC

- Knowledge Range (*KR*): concentrates on identifying answerable questions ("Answer") versus refusal situations ("Refusal") within the context of on-scene machine reading comprehension (MRC) dialogues.
- Style Compliance (*SC*): assesses whether the model can precisely generate role-specific responses such as "Answer," "No Answer," "Refusal," and "Attempt" in On-scene MRC Dialogues, without veering into narration.

- Nested Instruction-following (*NI*), Multi-turn Instruction-following (*MT*), Instruction Priority (*IP*): given complex higher-level constraints, these dimensions evaluate whether the model's responses meet the requirements of these constraints.

N Prompts for Reward Scoring

Below, we list the prompts used for reward scoring.

Reward Prompt for Basic Dialogue:

```
# An LLM is playing a specific character and answer a question about the given passages:
{conversations}
# Here is a reference answer.
{reference}
# The LLM generates the following response of the question:
\ "{response}"
# Ideally, the LLM should produce a response that clearly understands the user's intent and is
free of errors.
# Please refer to the reference answer, and evaluate the LLM's response. Below are the scoring
criteria.:
According to the criteria mentioned above, the better the answer, the closer it is to 1. Your score
should be a float between 0 and 1. (For instance, 0.5)
0 - The LLM's response contains a misunderstanding of the question, is only loosely relevant, or
contradicts the established context.
1 - The LLM's response accurately grasps the user's intent, is relevant. Respond using the first
person.
Output in dict format (do not output ``python, just output the dict): e.g. {{'think': [the reason of
your evaluation], 'score': [your score]}}
```

Figure 9: Reward prompt for Basic Dialogue

Reward Prompt for Persona Knowledge:

You will be given a conversation history and a question. You will then be given one response to the question.

Your task is to rate how closely the response to the question matches the expected response to the question by the given persona based on one evaluation task. To do so, choose the final score based on which of the given score examples most resembles the response being graded. You are to justify the reasoning behind the score you give the response. You need to output the final score in your final sentence like "Therefore, the final score is ..." Remember you are an ACCURATE, FAITHFUL, CRITICAL and FAIR judge.

Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed.

History:

{conversations}

Evaluation Criteria:

Ideally, the LLM should generate a response according to the character's ****knowledge range****.

1. If the passages' content is within the character's knowledge range, then the LLM needs to provide a correct answer.

2. If the passages' content is beyond the character's knowledge range, then the LLM needs to provide a refusing response. Except the LLM is required to give an answer in any case.

If the reference answers the question, then the content of the passages is within the character's knowledge range. Conversely, if it does not, it exceeds the character's knowledge range.

Please refer to the reference answer, and evaluate the LLM's response. Below are the scoring criteria.:

Score = 0: The LLM's response incorrectly answers the question that is beyond the character's knowledge range, or refuses to answer a question that is within the character's knowledge range.

Score = 0: Although the question is answered, the response is overly verbose for a simple question, or it mentions the persona's name itself, or includes too much information irrelevant to the question.

Score = 1: The response directly contradicts the attributes of the persona or introduces multiple new attributes not mentioned in the persona description. It may also incorrectly state that it is an "AI Model" or similar, showing a complete disregard for maintaining persona consistency.

Score = 2: The response is somewhat inconsistent with the persona, perhaps contradicting a key attribute or introducing an attribute not present in the description. The persona confidently answers a question regarding an attribute that is not explicitly mentioned in the persona description. There might be minor references to being an AI, affecting the persona's believability.

Score = 3: The response generally aligns with the persona but may include slight inaccuracies or unnecessary elaborations on the persona's attributes not explicitly mentioned in the description. No AI status is disclosed, but the alignment is not perfect.

Score = 4: The response is mostly consistent with the persona, correctly representing almost all attributes without contradiction or introducing irrelevant details. There is a minor deviation or ambiguity that doesn't majorly affect the overall consistency.

Score = 5: The response is fully consistent with the persona description, perfectly adhering to all mentioned attributes without deviation, omission, or the introduction of new, unrelated attributes. It does not acknowledge being an AI, maintaining complete immersion in the persona.

Score Examples:

{reference}

The LLM generates the following response of the question:

\n {response} \n

Your evaluation process is as follows: 1. Determine whether the reference refused to answer the question, which helps to judge whether the question should be refused. 2. If the question should be answered, then assess whether the provided answer truly addresses the question. 3. Additionally, evaluate whether the knowledge involved in the answer is relevant to the characteristics of the persona. Your evaluation should be extremely strict!

Output in dict format (do not output ``python, just output the dict): e.g. {{'think': [the reason of your evaluation], 'score': [your score]}}

Figure 10: Reward prompt for Persona Knowledge

Reward Prompt for Style Compliance:
You will be given a conversation history and a question. You will then be given one response to the question.

Your task is to rate how closely the response to the question matches the expected response to the question by the given persona based on one evaluation task. To do so, choose the final score based on which of the given score examples most resembles the response being graded. You are to justify the reasoning behind the score you give the response. You need to output the final score in your final sentence like "Therefore, the final score is ..." Remember you are an ACCURATE, FAITHFUL, CRITICAL and FAIR judge.

Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed.

History:
{conversations}

Evaluation Task Description:

The evaluation task of "linguistic habit" assesses the persona's adherence to its characteristically unique syntax, tone, and lingo, ensuring that these elements are consistently utilized throughout the persona's dialogue. This includes avoiding generic language patterns (such as "As a [persona]") and integrating specific idiomatic expressions, colloquialisms, or jargon that define the persona's distinctive verbal identity. The aim is to evaluate how effectively the persona maintains its linguistic uniqueness in various contexts.

Evaluation Criteria:

Score = 1: The response shows almost no alignment with the expected linguistic habits of the persona. It lacks any unique syntax, tone, or lingo specific to the persona, and could easily be attributed to any generic speaker without any distinct characteristics. There is a complete disregard for the persona's linguistic identity.

Score = 2: The response demonstrates minimal adherence to the persona's linguistic traits. There may be a slight attempt to use specific language patterns or tone, but these are either incorrectly applied or too sporadic to convey a true sense of the persona's unique verbal identity. The response predominantly uses generic language that does not match the persona's style.

Score = 3: The response includes a moderate level of persona-specific language, tone, and syntax. It shows an effort to incorporate the unique elements of the persona's linguistic habits, but these elements are not consistently maintained throughout the response. There may be occasional lapses into generic language or slight inconsistencies in tone.

Score = 4: The response is largely consistent with the persona's linguistic habits, using appropriate syntax, tone, and lingo throughout. There is a clear understanding of the persona's unique verbal identity, with only minor deviations or imperfections. The response shows a strong alignment with the persona's character but might lack a bit of finesse or have a single noticeable error.

Score = 5: The response perfectly captures the persona's linguistic habits, consistently using persona-specific syntax, tone, and idiomatic expressions. It fully adheres to the persona's unique verbal identity, integrating colloquialisms or jargon accurately and naturally. The response exemplifies how the persona would speak in virtually any given context, with no deviations from the persona's established linguistic style.

Here is a set of scores along with their corresponding responses.
{reference}

The LLM generates the following response of the question:
\"{response}\"

You need to examine the reference that corresponds to the score and determine how many points the current given response should receive.

Your evaluation should be extremely strict!
Output in dict format (do not output ``python, just output the dict): e.g. {{'think': [the reason of your evaluation], 'score': [your score]}}

Figure 11: Reward prompt for Style Compliance