MO-GRPO: Mitigating Reward Hacking of Group Relative Policy Optimization on Multi-Objective Problems

Yuki Ichihara[1] **Yuu Jinnai**[2] **Tetsuro Morimura**[2] **Mitsuki Sakamoto**[2]
Ryota Mitsuhashi[2] **Eiji Uchibe**[3]
[1]Nara Institute of Science and Technology [2]CyberAgent
[3]Advanced Telecommunications Research Institute International

Abstract

Group Relative Policy Optimization (GRPO) has been shown to be an effective algorithm when an accurate reward model is available. However, such a highly reliable reward model is not available in many real-world tasks. In this paper, we particularly focus on multi-objective settings, in which we identify that GRPO is vulnerable to reward hacking, optimizing only one of the objectives at the cost of the others. To address this issue, we propose MO-GRPO, an extension of GRPO with a simple normalization method to reweight the reward functions automatically according to the variances of their values. We first show analytically that MO-GRPO ensures that all reward functions contribute evenly to the loss function while preserving the order of preferences, eliminating the need for manual tuning of the reward functions' scales. Then, we evaluate MO-GRPO experimentally in four domains: (i) the multi-armed bandits problem, (ii) simulated control task (Mo-Gymnasium), (iii) machine translation tasks on the WMT benchmark (En-Ja, En-Zh), and (iv) instruction following task. MO-GRPO achieves stable learning by evenly distributing correlations among the components of rewards, outperforming GRPO, showing MO-GRPO to be a promising algorithm for multi-objective reinforcement learning problems.

1 Introduction

Reward hacking is a phenomenon in which an agent overfits to a misspecified reward model, failing to optimize for the true intended objective (Skalse et al., 2022; Gao et al., 2023; Rafailov et al., 2024). Recent work proposes Group Relative Policy Optimization (GRPO; Shao et al. 2024; Liu et al. 2025) to enhance the reasoning capability of Large Language Models (LLMs; Liu et al. 2024; Yang et al. 2025), demonstrating that the method can significantly improve the performance of the agent using a reward model with high accuracy. However, it is difficult to obtain high-accuracy reward models in many real-world tasks.

In this work, we evaluate GRPO's performance in a specific scenario where we only have access to under-specified reward models that do not accurately represent the task's objective on their own. In particular, we study a practical scenario where we specify the intended behavior of the agent using multiple reward models. The scenario involves a real-world machine translation task. The objective is to generate texts that are both (1) consistent with the content in the original language and (2) easy to read in the target language.

Our analysis shows that the loss function of the GRPO leads to optimizing the reward functions with higher variances while ignoring the lower ones, which may result in an undesirable policy. We empirically evaluate GRPO on multi-objective reinforcement learning problems, where we also observe reward hacking behavior of GRPO that it tends to ignore rewards with lower variances and only optimizes the ones with higher variances, resulting in an unintended behavior (e.g., hacking the readability objective while ignoring the translation consistency; Table 1).

To resolve this problem, we propose GRPO with a simple automated normalization method to the objectives, which we call **MO-GRPO (Multi-Objective Group Relative Policy Optimization)**. MO-GRPO normalizes the advantage functions for each objective so that their variances are scaled evenly. This ensures that all reward functions contribute equally to updating the policy, regardless of their variance scales (Theorem 2). In this way, it prevents any objectives from being ignored in the training process, mitigating the reward hacking behavior of GRPO on multiple objectives. Our normalization technique maintains the origi-

Instruction	**Translate the following English into easily readable Japanese.**\nOver the past decade, our lives have changed through technology, with many working from home, ...	jReadability↑	BLEURT↑
GRPO	Over the past ten years, our lives have changed a lot because of technology. ...	**0.99**	0.57
MO-GRPO	過去の10年で、技術の進化により、多くの人がホームワークから仕事をしているようになり... (Translation: In the past decade, technological advances have enabled many people to work from home work...)	0.40	0.69

Table 1: (Machine translation) Generation examples of GRPO and MO-GRPO by Llama (Llama-3.2-3B-Instruct). **GRPO optimizes only the Japanese readability score (jReadability) by avoiding using difficult Japanese words, eventually stops using any Japanese characters**, ignoring the translation accuracy score (BLEURT), resulting in generating non-Japanese text, which defeats the purpose of the translation. On the other hand, MO-GRPO evenly optimizes both objectives, achieving improvement on both objectives as intended. Generation examples from other LLMs and the results of outputs during training are shown in Appendix D.

nal preference ordering under positive affine transformations of reward scales, even in cases where GRPO fails to preserve this ordering.

We evaluate MO-GRPO experimentally on multi-armed bandit, simulated control (Felten et al., 2023), machine translation (Kocmi et al., 2024) problems, and instruction following task with multiple objectives. The result shows that MO-GRPO successfully mitigates reward hacking in the problem settings where GRPO incurs reward hacking (Table 1), resulting in a policy that is desirable for the given task.

2 Related Works

Reward hacking is one of the known problems in reinforcement learning (Amodei et al., 2016; Ziegler et al., 2020; Stiennon et al., 2020; Skalse et al., 2022; Gao et al., 2023). In the context of Large Language Models (LLMs), reward hacking often becomes a problem in the alignment process (Rafailov et al., 2024) where the LLMs are optimized to generate outputs that maximize the score of a reward model trained using human feedback (e.g., RLHF; (Ziegler et al., 2020; Stiennon et al., 2020)). Prior work shows that if the reward model is inaccurate, the LLM can learn to generate sentences that increase the score from the reward model, even if it does not improve the true intended objective of the training. For example, it might produce outputs that are very long, overly polite, or stylistically pleasing to the reward model, even if they are factually incorrect or unhelpful, because such attributes are spuriously correlated with high reward scores (Gao et al., 2023). This over-optimization on a proxy reward can lead to a decrease in the true objective and alignment of the model's outputs (Pan et al., 2022; Gleave et al., 2020; Kim et al., 2025). There are several fine-tuning methods for multi-objective problems that can be used to solve this problem (Zhou et al., 2024; Li et al., 2025). However, they face some problems. The problem of Zhou et al. (2024) is that, as the ratio of conflicts in the training data increase, the Pareto Fronts gradually move downwards, showing significant performance decreases. Li et al. (2025) can resolve this problem, but the proposed method involves response sampling, refinement, filtering, and fine-tuning, which could introduce computational overhead compared to simpler methods.

3 Group Relative Policy Optimization (GRPO)

GRPO is a reinforcement learning algorithm (Shao et al., 2024) which is usually used for online and on-policy learning. For a given state, a policy generates multiple outputs and learns to generate outputs with higher relative reward scores compared to the rest of the outputs. p_Q is the distribution over the initial state (prompt) ($q \sim p_Q$). The policy $\pi_\theta(\cdot \mid q)$ outputs the action (sentence) o_g based on the initial state q from the action space. Formally, let R_i be the i-th reward function, which is a mapping from a prompt-output pair to a scalar

value, and assume that there are K reward functions. For each prompt q, GRPO samples a group of outputs $\mathbf{o} = \{o_1, o_2, ..., o_G\}$ from the reference policy $\pi_{\theta_{\text{ref}}}$ and then optimizes the policy model by maximizing the following objective:

$$\mathcal{J}(\pi_\theta) = \mathbb{E}\Bigg[\sum_{g=1}^{G} \frac{1}{G}\frac{1}{|o_g|}\frac{\pi_\theta(o_g \mid q)}{\pi_{\theta_{\text{ref}}}(o_g \mid q)} A_g - \beta \mathrm{KL}(\pi_\theta, \pi_{\theta_{\text{ref}}})\Bigg], \tag{1}$$

where β is a hyperparameter, and KL is Kullback–Leibler (KL) divergence. Since we omit symbols such as the threshold ϵ and min operation for simplicity, we note formal expressions in the Appendix B.

A_g represents the normalized advantage value of the sentence o_g using K reward models:

$$A_g = \frac{\sum_{i=1}^{K} R_i(q, o_g) - \text{mean}_{\mathbf{o}}\left(\sum_{i=1}^{K} R_i(q, \mathbf{o})\right)}{\text{std}_{\mathbf{o}}\left(\sum_{i=1}^{K} R_i(q, \mathbf{o})\right)}. \tag{2}$$

The advantage value A_g is computed without normalizing the scale of the reward functions (Eq. 2). Consequently, when rewards differ in scale or variance, the advantage value can be dominated by the value from a high variance reward function.

Theorem 1 (Correlation between reward function and advantage function with GRPO)**.** *Assume the $G \to \infty$. The correlation coefficient between an individual reward function R_i and the advantage A_g is the ratio of R_i's standard deviation σ_i to the standard deviation of the total reward σ.*

$$\text{Corr}(R_i(q, o_g), A_g) = \frac{\sigma_i^2 + X}{\sigma\sigma_i} \tag{3}$$

where $X = \sum_{j \neq i} \text{Cov}(R_i, R_j)$, $\text{Cov}(\cdot, \cdot)$ is covariance.

The proof is in Appendix E.1.

Theorem 1 shows that the advantage function in GRPO is more strongly correlated with reward components that exhibit higher variance. This shows that GRPO learns to optimize reward functions with higher variances than the lower ones and may lead to unintended behavior, which we show empirically in Section 5.

4 Multi-Objective GRPO (MO-GRPO)

To solve the problem that GRPO is affected by the scale of the variance of the reward functions, we propose Multi-Objective GRPO (MO-GRPO). By computing a separate advantage function for each reward, our framework adjusts for differences in reward variance and enables more stable learning.

$$\mathcal{J}(\pi_\theta) = \mathbb{E}\Bigg[\sum_{g=1}^{G} \frac{1}{G}\frac{1}{|o_g|}\frac{\pi_\theta(o_g \mid q)}{\pi_{\theta_{\text{ref}}}(o_g \mid q)} A_g^{\text{MO}} - \beta \mathrm{KL}(\pi_\theta, \pi_{\theta_{\text{ref}}})\Bigg], \tag{4}$$

where A_g^{MO} is defined as follows:

$$A_g^{\text{MO}} = \sum_{i=1}^{K} \frac{R_i(q, o_g) - \text{mean}_{\mathbf{o}}\left(R_i(q, \mathbf{o})\right)}{\text{std}_{\mathbf{o}}\left(R_i(q, \mathbf{o})\right)}. \tag{5}$$

Note that **MO-GRPO rescales the reward *individually*, then aggregating over the reward functions**, whereas **vanilla GRPO rescales it after all the reward values are aggregated into a single value (Equation 2)**. Thus, MO-GRPO ensures a consistent correlation between each advantage function and its corresponding reward function.

Theorem 2 (Correlation between a reward function and advantage function with MO-GRPO)**.** *Assume that the number of samples $G \to \infty$. The correlation of the advantage functions A_g^{MO} with each reward function R_i for any o_g remains constant.*

$$\text{Corr}(R_i(q, o_g), A_g^{\text{MO}}) = \frac{1 + Z}{\sqrt{K + Y}} \tag{6}$$

where $Y = \sum_{j=1}^{K} \sum_{l \neq j} \frac{\text{Cov}(R_l, R_j)}{\sigma_l \sigma_j}$ and $Z = \sum_{j \neq i} \frac{\text{Cov}(R_i, R_j)}{\sigma_i \sigma_j}$, $\text{Cov}(\cdot, \cdot)$ is covariance.

The proof is in Appendix E.2.

Theorem 2 shows that the advantage function in MO-GRPO is roughly equal to $\frac{1}{\sqrt{K}}$ for every reward function R_i, with some effect of correlation between the reward functions. Specifically, if all the reward functions are uncorrelated with each other, then the correlation of the reward and the advantage is exactly $\frac{1}{\sqrt{K}}$ for all the reward functions:

Corollary 1 (Correlation between a reward function and advantage function with MO-GRPO under certain assumptions)**.** *Assume that the K reward functions R_i are mutually uncorrelated and the number of samples $G \to \infty$. The correlation*

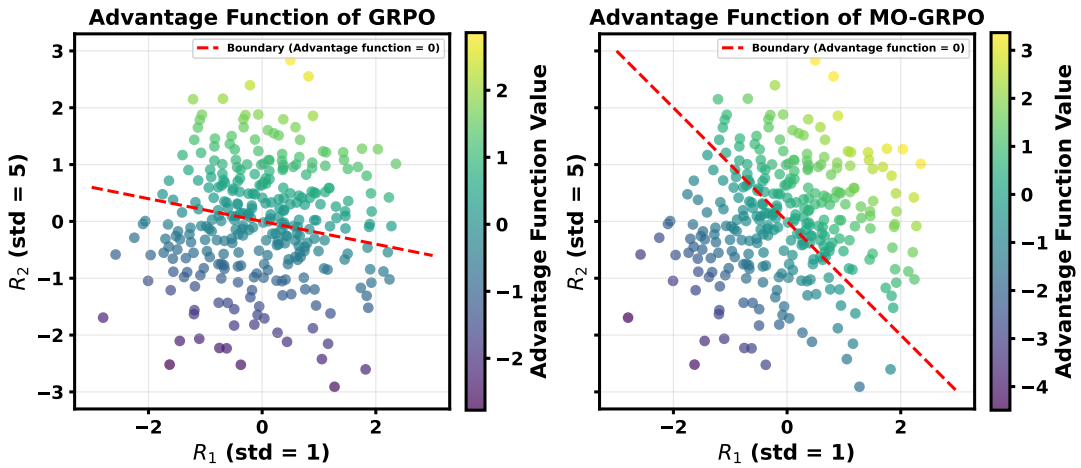

Figure 1: (Simulated experiment) Comparison of the advantage values of GRPO and MO-GRPO on a toy example with two reward functions with different sizes of variances (1 and 5). The advantage values of GRPO (left figure) are dominated by the high variation reward (R_2), indicating that the algorithm is sensitive to the relative scales of the rewards. In contrast, the advantage values of MO-GRPO (right figure) are invariant with the scale of the reward models, which shows that MO-GRPO is an easy-to-use algorithm for multi-objective learning tasks that does not require manual tuning of the reward models to avoid reward hacking.

of the advantage functions A_g^{MO} with each reward function R_i for any o_g remains constant.

$$\text{Corr}(R_i(q, o_g), A_g^{\text{MO}}) = \frac{1}{\sqrt{K}} \tag{7}$$

The proof follows immediately from Theorem 2. The result shows that the reward functions of the MO-GRPO have roughly the same amount of influence on the policy update regardless of their variances. Thus, MO-GRPO does not ignore reward functions with lower variances, which could lead to unintended behavior. We show this property empirically in Section 5.

Example. Here, we demonstrate how MO-GRPO preserves the sensitivity of each reward function within the advantage function. We consider two reward functions and three outputs $R_1 : [0.1, 0.5, 0.9]$ and $R_2 : [0.15, 0.13, 0.05]$. A case of GRPO, reward mean is 0.61 and standard deviation is 0.29, therefore, the advantage functions are $[-1.38, 0.43, \mathbf{0.95}]$. On the other hand a case of MO-GRPO, reward means are $[0.5, 0.11]$ and standard deviation: $[0.33, 0.04]$, therefore advantage function are $[-1.22+0.93, 0.0+0.46, 1.22-1.39]$. From this, it can be said that MO-GRPO successfully reflects the superiority of R_2 in the advantage function. This can be demonstrated by the following Theorem 2.

Simulated experiment. Figure 1 shows the comparison of the advantage functions of GRPO (Eq. 2) and MO-GRPO (Eq. 5) with two reward functions with low and high variances. The reward functions return a value sampled from a normal distribution $R_1 \sim \mathcal{N}\left(1, 1^2\right)$ and $R_2 \sim \mathcal{N}\left(1, 5^2\right)$. The advantage value by GRPO is significantly influenced by R_2 while the effect of R_1 is negligible. This motivates the agent to maximize the value of R_2 even at the cost of losing R_1. Conversely, the advantage value calculated by MO-GRPO successfully considers both reward functions, even when their variances differ significantly. This indicates that, when using MO-GRPO, one does not need to manually adjust the scale of the reward models for a given task to prevent the agent from performing unbalanced optimization.

Invariance to positive affine transformation. An additional advantage of MO-GRPO is its invariance under positive affine transformations of reward functions.

Proposition 1 (Affine Invariance of MO-GRPO Advantage)**.** *Let o_a and o_b be two possible outputs, and let $\mathcal{R} = \{R_i\}_{i=1}^K$ be a set of reward functions. Consider a transformed set $\mathcal{R}' = \{R'_i = a_i R_i + b_i\}_{i=1}^K$ with $a_i > 0$. Then, the preference ordering induced by MO-GRPO is invariant under*

such positive affine transformations:

$$A_a^{\text{MO}} \geq A_b^{\text{MO}} \iff A_a^{\text{MO}'} \geq A_b^{\text{MO}'}$$

where $A_a^{\text{MO}'} = \sum_{i=1}^{K} \frac{R_i'(q,o_a) - \text{mean}_{\mathbf{o}}\left(R_i'(q,\mathbf{o})\right)}{\text{std}_{\mathbf{o}}\left(R_i'(q,\mathbf{o})\right)}$.

The proof of Proposition 1 is in Appendix E.3. Meanwhile, Theorem 2 shows that MO-GRPO does not require engineering efforts to normalize the scale of the reward functions *relative* to other reward functions, Proposition 1 shows that it does not need to normalize the *absolute* scale of the reward functions. This makes MO-GRPO a practically useful algorithm for real-world problems where the scale of the reward models is unclear (e.g., neural models) or instance-dependent. Conversely, this property does not hold with GRPO.

Proposition 2. *The preference ordering induced by GRPO (and Dr .GRPO) is not invariant under positive affine transformations.*

The proof is in Appendix E.4.

Summary. Unlike GRPO (Theorem 1), the advantage function of MO-GRPO is built keeping the correlation with each reward function constant (Theorem 2). In addition, it is invariant with the rescaling of the reward functions with positive affine transformation (Proposition 1).

These properties make MO-GRPO an easy-to-use algorithm. It can use off-the-shelf reward functions without requiring manual tuning of the reward values to fit them to target tasks.

5 Experiment

We conduct experiments on four tasks: (1) multi-armed bandit, (2) simulated control, (3) machine translation, and (4) instruction following task. We compare three methods: GRPO, Dr. GRPO, and MO-GRPO.

5.1 Multi-Armed Bandit

We first conduct experiments on a simple multi-armed bandit environment to observe the behavior of GRPO and MO-GRPO in a controlled environment. We set the number of arms (actions) k to 50, and there are three stochastic reward functions R_1, R_2, and R_3. The episode length is fixed to 5000 steps. The expected return of the arm μ_k is chosen at random from a normal distribution of $\mathcal{N}(0, 1)$ at the beginning of the episode and is fixed throughout the episode. The three reward functions output the reward value of μ_k plus additional stochastic noise to it as follows:

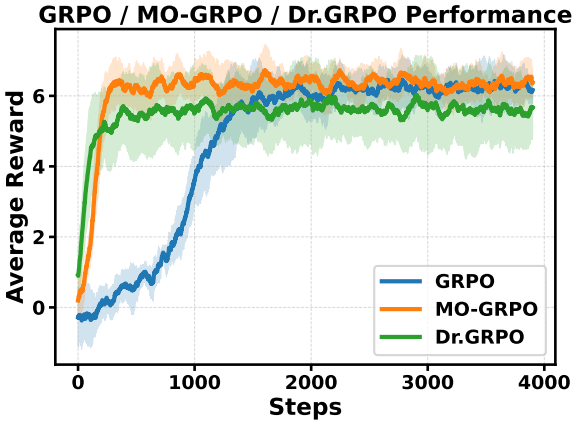

Figure 2: (Multi-armed bandit) This figure illustrates the average rewards obtained by the sum of the three reward functions: GRPO, MO-GRPO, and Dr. GRPO. As the figure shows, MO-GRPO finds a better policy faster than GRPO and Dr. GRPO.

$$R_1(k) \sim \mathcal{N}(\mu_k, 10^2)$$
$$R_2(k) \sim \mathcal{N}(\mu_k, 1^2) - 0.1R_1(k)$$
$$R_3(k) \sim \mathcal{N}(\mu_k, 0.1^2)$$

$R_1(k)$ outputs a value sampled from a normal distribution of mean μ_k and standard deviation of 10. $R_2(k)$ is a sum of a value sampled from a normal distribution of mean μ_k and standard deviation of 1, minus the value of $0.1 \times R_1$. Thus, R_2 is negatively correlated with R_1, making it harder to learn. R_3 is a value sampled from a normal distribution of mean μ and standard deviation of 0.1. These reward functions are designed to create a challenging optimization landscape where a high-variance reward function (R_1) could dominate the learning signal, potentially overshadowing the gradients from lower-variance and/or negatively correlated reward functions (R_2 and R_3).

We set the number of actions GRPO, Dr. GRPO, and MO-GRPO samples G to 8, and we use a neural network with 3 hidden layers for policy. We conduct experiments with five different random seeds.

Figure 2 shows the comparison of the three algorithms on the sum of the three reward functions. The result shows that MO-GRPO achieves a better policy faster than the others. Figure 3 shows the breakdown of the three reward functions, showing

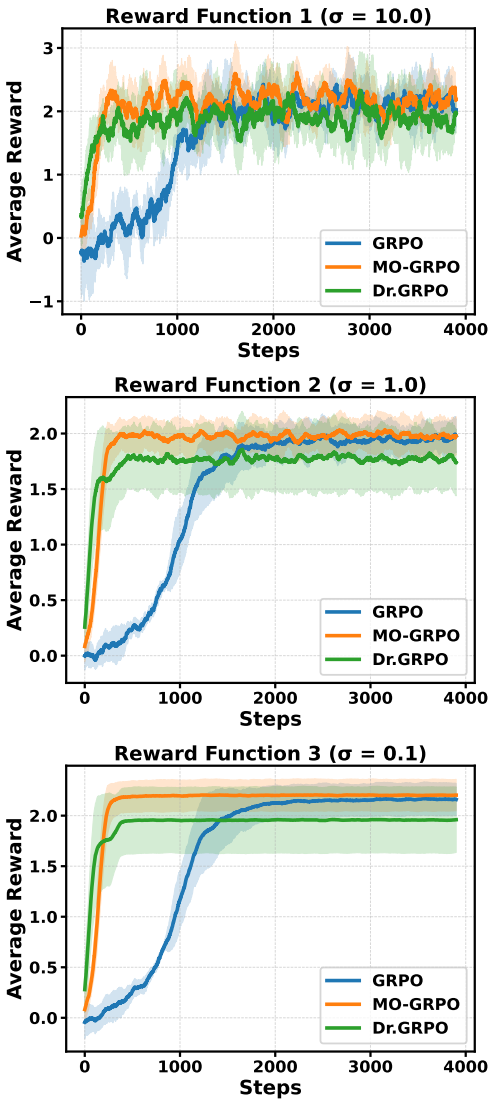

Figure 3: (Multi-arm bandit) Comparison of the three reward functions with varying variances (10, 1, and 0.1) obtained by GRPO, Dr. GRPO, and MO-GRPO. While GRPO and Dr. GRPO fail or are slow to learn the reward functions with lower variances (R_2 and R_3), MO-GRPO successfully optimizes all three reward functions regardless of the scale of the variances.

that GRPO and Dr. GRPO fail to learn low variance reward functions (R_2 and R_3), resulting in suboptimal policies.

The experimental result on a multi-armed bandit problem shows that MO-GRPO is a promising approach for tasks where the reward functions have different scales of variance.

5.2 Simulated Control

To evaluate the method on a simulated control task, we utilize the MO-Reacher (`mo-reacher-v5`) control benchmark from the `mo_gymnasium` framework (Felten et al., 2023), as in Figure 4. In this task, a policy controls a two-joint robotic arm to simultaneously reach four distinct targets within 50 steps, which is an episode. The state is represented by a 6-dimensional vector composed of the sin and cos of the two joint angles and their angular velocities, while the policy selects from a discrete action space of nine. The objective of the task is to maximize the sum of the 4 reward functions, where each component R_i is a function of the squared Euclidean distance from the arm position to the target position: $R_i = 1 - 4 \cdot \left\| p_{\text{arm}} - p_{\text{target},i} \right\|_2^2$. We conduct this experiment with five different random seeds.

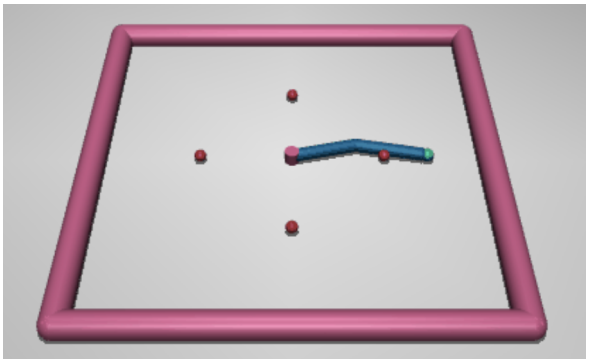

Figure 4: Simulated control task we use for the experiment. Two-joint arms with a 6-state vector (sin, cos of joint angles and their angular velocities) select among 9 discrete actions to reach four targets within a 50-step episode. Each reward function is defined as $R_i = 1 - 4\|p_{\text{arm}} - p_{\text{target},i}\|_2^2$. The optimal control in this environment is to keep swinging the arm at a constant speed.

Table 2 shows the average total reward obtained per step. As can be seen, MO-GRPO can obtain a higher reward value per step than GRPO, demonstrating its effectiveness, even when applied to conventional reinforcement learning tasks. For a supplementary explanation, both methods show large values for R_1 and R_3 because GRPO and MO-GRPO learned policies that emphasized rewards for R_1 and R_3 once each, while producing very negative values for R_2 and R_4 out of five seed iterations. This can also be confirmed by looking at the standard deviations of R_2 and R_4.

In the ablation study, we modified the environment by setting the standard deviation of R_1 to 2. This verifies how MO-RPO behaves when encoun-

Method	Total Reward ↑	Each Reward ↑			
		R_1	R_2	R_3	R_4
GRPO	1.29±0.24	0.39±0.11	0.24±0.15	0.44±0.08	0.20±0.17
Dr. GRPO	1.10±0.20	0.33±0.06	0.22±0.05	0.32±0.07	0.22±0.06
MO-GRPO	**1.48±0.26**	**0.45±0.04**	**0.29±0.18**	**0.46±0.08**	**0.28±0.18**

Table 2: (Simulated control) MO-Reacher results. The total reward shows the sum of the four reward values (R_1, R_2, R_3, and R_4) per step. The optimal policy achieves the total reward of around 1.76.

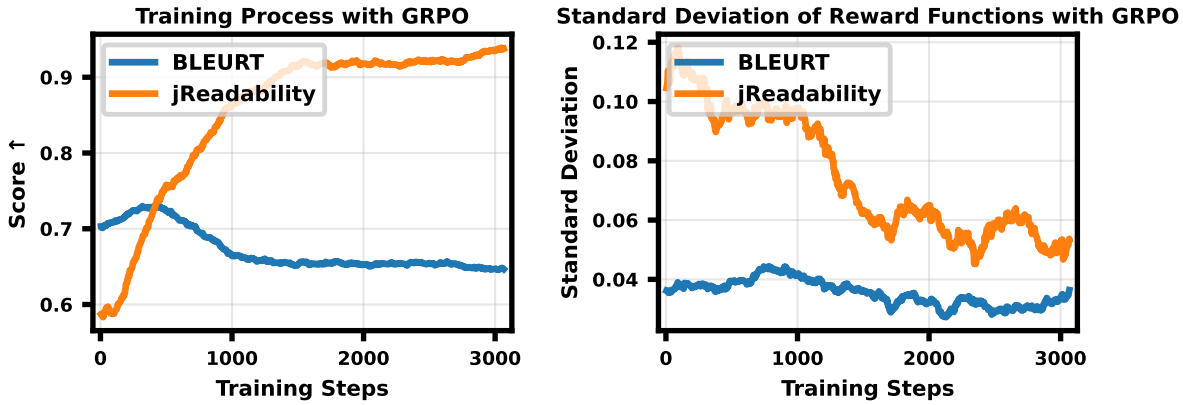

Figure 5: (Machine translation) The training process of GRPO on the WMT En-Ja dataset uses BLEURT and jReadability as the reward functions by Sarashina (sarashina2.2-3b-instruct-v0.1). As the results show, GRPO overfits jReadability at the expense of BLEURT performance. As the results show, the standard deviation of jReadability is always more than BLEURT. As shown in Appendix K, the same phenomenon is observed in other LLMs.

tering reward functions with high variance.

Table 3 shows the average reward values for each reward function per step with the standard deviation of R_1, set to 2. MO-GRPO has lower values than GRPO for R_1 (high-standard-deviation reward function), but improves upon GRPO for the others. This suggests that, unlike GRPO, MO-GRPO can learn stably even when the standard deviations of each reward function diverge. It can be seen that Dr. GRPO exhibits behavior similar to that of GRPO in this setting. The behavior of GRPO and MO-GRPO is shown in Appendix I.

5.3 Machine Translation

We evaluate the performance of MO-GRPO on machine translation with two objective functions, translation accuracy and readability. Readability is one of the important objectives in real-world text generation tasks (Hasebe and Lee, 2015; Trokhymovych et al., 2024). It measures the accessibility of the text for a diverse audience, including children and non-native speakers, and is critical for communicating vital information during emergencies such as natural disasters.

We conduct experiments on English to Japanese (En-Ja) and English to Chinese (En-Zh). We use the WMT-21, WMT-22, and WMT-23 datasets for training (Akhbardeh et al., 2021; Freitag et al., 2022, 2023), and evaluate on the WMT-24 test set (Kocmi et al., 2024).

First, we perform the En-Ja translation task in WMT datasets using Sarashina (`sarashina2.2-3b-instruct-v0.1`), Qwen (`Qwen2.5-3B-Instruct`) (Yang et al., 2025), and Llama (`Llama-3.2-3B-Instruct`) (Grattafiori et al., 2024) as the base models. For the reward (objective) functions, we adopt (i) BLEURT (Sellam et al., 2020) and (ii) jReadability (Hasebe and Lee, 2015) to measure readability in Japanese. To evaluate the overall generation quality, we use LLM-as-a-Judge (Zheng et al., 2023) with GPT-4o-mini (GPT-Eval) so that both the translation accuracy and readability are considered.

Table 4 shows that, compared to the base model score, GRPO achieved a high jReadability score

Method	Total Reward ↑	Each Reward ↑			
		R_1	R_2	R_3	R_4
GRPO	1.14±0.21	**0.37±0.05**	0.21±0.06	0.35±0.10	0.21±0.05
Dr. GRPO	1.10±0.16	0.35±0.02	0.20±0.07	0.34±0.09	0.21±0.04
MO-GRPO	**1.40±0.25**	0.36±0.09	**0.33±0.05**	**0.40±0.05**	**0.30±0.07**

Table 3: (Simulated control) MO-Reacher by setting the standard deviation of only the first reward function R_1 to 2 results. The total reward shows the sum of the four reward values (R_1, R_2, R_3, and R_4) per step. The optimal policy achieves the total reward of around 1.76.

but at the cost of degrading the BLEURT score. This result leads to the worst win rate score against the base model in three methods. In contrast, MO-GRPO almost successfully improved both metrics compared to the base model's score, achieving in BLEURT and jReadability scores, preventing overfitting to jReadability, and MO-GRPO also achieves the highest win rate score. For supplementary, Dr. GRPO also shows higher values for both metrics compared to the base model, but not as high as MO-GRPO with respect to GPT-Eval (Table 11).[1]

In detail, MO-GRPO with Sarashina achieves BLEURT score of 0.69. For comparison, when Sarashina is trained with GRPO solely on BLEURT, the score reached 0.70. This close score suggests that MO-GRPO effectively learns to optimize BLEURT without sacrificing other objectives. Furthermore, as shown in the output examples (Table 10), GRPO exhibits behavior analogous to reward hacking on jReadability; on the other hand, it is not observed with MO-GRPO, and the training process of MO-GRPO with Sarashina is shown in Figure 6, which suggests MO-GRPO avoids overfitting of jReadability and prevents degradation of BLEURT, unlike GRPO (Figure 5).

Next, we examine the details of the results from other language models such as Qwen and Llama. The relatively limited improvement of MO-GRPO in Qwen's experiments is likely attributable to Qwen not being a language model specialized for Japanese. However, Qwen outputs (Table 9) are also confirmed reward hacking behavior from GRPO (such behavior is also not observed with MO-GRPO). Llama outputs (Table 1) show that GRPO engaged in reward hacking by

[1]Dr. GRPO in this experiment is implemented using trl=0.16.1. This version of Dr. GRPO has no exclusions regarding sentence length normalization, only in the form of removing the standard deviation of the advantage function.

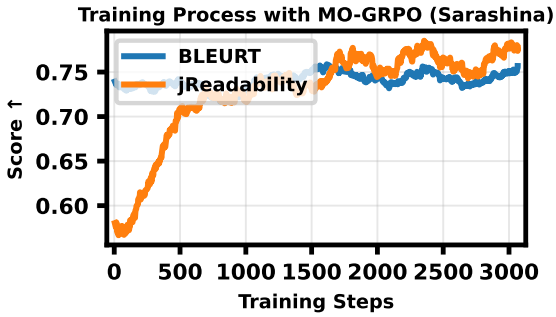

Figure 6: (Machine translation) The training process of MO-GRPO on the WMT En-Ja dataset uses BLEURT and jReadability as the reward functions by Sarashina. Unlike GRPO (Figure 5), the figure shows MO-GRPO avoids overfitting of jReadability and prevents deterioration of BLEURT. We also see stability in the results in other LLMs (Appendix K).

outputting English text instead of a Japanese translation, thereby improving the jReadability. This phenomenon again did not occur with MO-GRPO. As discussed in Appendix K, GRPO with Llama exhibits the strongest overfitting to jReadability, which can be explained by the higher variance of this metric under Llama.

Second, we conduct En-Zh translation task in WMT datasets using Qwen and Llama as base models. For the reward functions, we adopt (i) BLEURT and (ii) TRank (Trokhymovych et al., 2024), which can evaluate the readability of text across multiple languages, including Chinese. Since TRank scores are higher for more difficult texts, multiply the score by -1 in this experiment setting during the training (Table 4 shows the true TRank score, i.e., the score obtained without applying the -1 multiplication during evaluation.).

As shown in Table 4, MO-GRPO also appropriately treats the two reward functions across Qwen and Llama, similar to the En-Ja task. There-

Base Model	Method	WMT24 (En–Ja)		
		BLEURT† ↑	jReadability† ↑	GPT-Eval⋆ ↑
Sarashina	Base Model	0.66	0.70	50.0%
	GRPO	0.62	**0.86**	66.0%
	Dr. GRPO	0.67	0.73	68.4%
	MO-GRPO (ours)	**0.69**	0.76	**76.8%**
Qwen	Base Model	**0.67**	0.66	50.0%
	GRPO	0.65	**0.74**	53.0%
	Dr. GRPO	**0.67**	0.66	69.6%
	MO-GRPO (ours)	**0.67**	0.67	**88.8%**
Llama	Base Model	0.65	0.67	50.0%
	GRPO	0.60	**0.90**	35.6%
	Dr. GRPO	0.63	0.77	42.6%
	MO-GRPO (ours)	**0.66**	0.69	**68.8%**
		WMT24 (En–Zh)		
Base Model	**Method**	**BLEURT†** ↑	**TRank†** ↓	**GPT-Eval⋆** ↑
Qwen	Base Model	**0.73**	-2.73	50.0%
	GRPO	0.71	**-3.20**	53.9%
	Dr. GRPO	**0.73**	-2.98	62.4%
	MO-GRPO (ours)	**0.73**	-2.85	**68.1%**
Llama	Base Model	0.69	-2.44	50.0%
	GRPO	0.62	**-2.98**	33.2%
	Dr. GRPO	0.60	-3.15	28.3%
	MO-GRPO (ours)	**0.71**	-2.55	**71.7%**

Table 4: (Machine translation) Translation quality on WMT24 (higher is better). †**BLEURT**, **jReadability**, and **TRank** are training objectives and thus susceptible to over-fitting. ⋆**GPT-Eval** (against Base Model) is not optimized during training; we therefore regard it as the *primary metric*. Across all three base models, our MO-GRPO improves GPT-Eval while avoiding excessive optimization of the training-objective metrics.

fore, it consistently achieves higher GPT-Eval win rates than GRPO and Dr. GRPO. GPT-Eval indicates that GRPO and Dr. GRPO are improving with Qwen but exhibit clear reward hacking with Llama. Both methods overfit to TRank at the expense of BLEURT. Interestingly, the trained models output non-Chinese text even though the task is Chinese translation (similarly to what we observed in En-Ja translation task in Table 1).

Table 5 further quantifies this phenomenon: using langdetect,[2]. We find that GRPO and Dr. GRPO frequently produce non-Chinese generations, exploiting TRank and this overly yields lower GPT-Eval scores than the base model.

Ablation study. Instead of using MO-GRPO, one may solve the reward hacking by patching the reward function so that it cannot be hacked. We improve the TRank reward function by giving a huge penalty (score=10) if the generated output is identified as non-Chinese by langdetect. In this way, we can prevent the model to learn to generate non-Chinese texts. Table 5 shows that adding a penalty to TRank reduces the probability of non-Chinese outputs (**Non-Chinese**) for all methods; MO-GRPO still has the lowest probability. Additionally, Table 6 shows the TRank with penalties under the same experimental settings as Table 4. This shows that MO-GRPO outperforms other methods in terms of GPT-Eval, and in this setting as well, GRPO and Dr. GRPO are trained with a focus on TRank penalties, which fluctuate more than BLEURT. This shows that MO-GRPO is on par with GRPO even if the reward functions are reasonably designed (e.g., problem settings in which GRPO achieves improvement).

[2]https://pypi.org/project/langdetect/

Method	Non-Chinese (%) (w/o Penalty) ↓	Non-Chinese (%) (w/ Penalty) ↓
Base Model (Llama)	14.7%	14.7%
GRPO	68.7%	1.2%
Dr. GRPO	70.7%	1.2%
MO-GRPO	5.6%	**0.6%**

Table 5: The probability of non-Chinese outputs (**Non-Chinese**) in machine translation. The reference set contains 851 Chinese sentences (by langdetect). **Non-Chinese** $= 1 - \#\text{Chinese}/851$. **w/o Penalty**: TRank only. **w/ Penalty**: TRank with penalty for non-Chinese outputs. MO-GRPO consistently maintains proper outputs under both settings.

Method	BLEURT ↑	TRank w/ Penalty ↓	GPT-Eval ↑
Base Model	0.69	1.39	50%
GRPO	0.70	-0.66	71.5%
Dr. GRPO	0.70	-0.64	69.6%
MO-GRPO	0.71	-0.47	**74.0%**

Table 6: (Machine translation) Translation quality on WMT24 En-Zh (higher is better) with Llama. **TRank w/ Penalty** penalty non-Chinese outputs. MO-GRPO improves GPT-Eval while avoiding excessive optimization of the training-objective metrics.

5.4 Instruction Following Task

In this section, we conduct an experiment in AlpacaFarm (training dataset: `tatsu-lab/alpaca`, eval dataset: `tatsu-lab/alpaca_eval`) (Dubois et al., 2023) to evaluate the performance of MO-GRPO for the generic instruction following task using Qwen and Llama as base models. We use RM-Mistral-7B (Dong et al., 2023) and the Length reward function. The Length reward function (R_{Len}) gives a higher reward on the outputs closer to the length of the reference text so that it mitigates the length bias problem (Shen et al., 2023; Singhal et al., 2024). Length reward function is defined as follows:

$$R_{\text{Len}} = \begin{cases} \dfrac{L}{0.9L_{\text{ref}}}, & L < 0.9L_{\text{ref}}, \\ 1, & 0.9L_{\text{ref}} \leq L \leq 1.1L_{\text{ref}}, \\ \dfrac{1.1L_{\text{ref}}}{L}, & L > 1.1L_{\text{ref}}. \end{cases}$$

where L_{ref} is the reference text length, L is the output text length. Given that RM-Mistral-7B tends to prefer longer outputs (Shen et al., 2023; Singhal et al., 2024), the Length reward function is an adversarial objective to it, making the optimization more challenging.

Base Model	Method	AlpacaFarm RM-Mistral-7B ↑	AlpacaFarm Length ↑
Qwen	Base Model	5.55	0.42
	GRPO	5.81	0.36
	Dr. GRPO	6.24	0.34
	MO-GRPO (ours)	5.51	0.44
Llama	Base Model	5.26	0.42
	GRPO	5.56	0.37
	Dr. GRPO	5.90	0.34
	MO-GRPO (ours)	5.28	0.42

Table 7: (AlpacaFarm) Since RM-Mistral and Length have conflicting objectives, the correct answer here is to prevent it from being derived from the base model. GRPO and Dr. GRPO have learned to prioritize RM-Mistral, resulting in a significant sacrifice of Length, but MO-GRPO retains both reward functions almost entirely.

Table 7 shows that GRPO and Dr. GRPO optimize RM-Mistral while decreasing the Length of both Llama and Qwen. In contrast, MO-GRPO attempts to maintain the values of both rewards. In such adversarial cases where both reward functions are important, the optimal behavior is to remain close to the base model, such as MO-GRPO.

6 Conclusion

We conducted an investigation into the theoretical and empirical properties of handling multiple reward functions with GRPO. Our analysis revealed a previously unreported vulnerability. The advantage function of GRPO is biased toward reward functions with high variance. This makes the algorithm susceptible to reward-hacking behaviors in multi-objective settings. To address this weakness, we proposed Multi-Objective GRPO (MO-GRPO), an extension of GRPO that uses a simple normalization method to automatically reweight reward functions according to their value variances. MO-GRPO treats each reward function value equitably while preserving preference orderings under rescalings. Comprehensive experiments confirmed the practical benefits of this mechanism. We experimentally evaluate MO-GRPO in four domains: (i) the multi-armed bandits problem, (ii) the simulated control task (Mo-Gymnasium), (iii) machine translation tasks on the WMT benchmark (En-Ja, En-Zh), and (iv) instruction following task. MO-GRPO consistently

avoids reward hacking and shows improvements in task-specific metrics (e.g., BLEURT, jReadability) and learning stability.

References

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 Conference on Machine Translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online. Association for Computational Linguistics.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. Concrete Problems in AI Safety. *arXiv preprint arXiv:1606.06565*.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, KaShun SHUM, and Tong Zhang. 2023. RAFT: Reward ranked Finetuning for Generative Foundation Model Alignment. *Transactions on Machine Learning Research*.

Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. 2023. AlpacaFarm: A Simulation Framework for Methods that Learn from Human Feedback. In *Advances in Neural Information Processing Systems*, volume 36, pages 30039–30069. Curran Associates, Inc.

Florian Felten, Lucas N. Alegre, Ann Nowe, Ana Bazzan, El Ghazali Talbi, Grégoire Danoy, and Bruno C. da Silva. 2023. A toolkit for reliable benchmarking and research in multi-objective reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pages 23671–23700. Curran Associates, Inc.

Xiao Feng, Bo Han, Zhanke Zhou, Jiaqi Fan, Jiangchao Yao, Ka Ho Li, Dahai Yu, and Michael Ng. 2025. DyPO: Dynamic Policy Optimization for Multi-Turn Interactive Reasoning. In *ICML 2025 Workshop on Programmatic Representations for Agent Learning*.

Markus Freitag, Nitika Mathur, Chi-kiu Lo, Eleftherios Avramidis, Ricardo Rei, Brian Thompson, Tom Kocmi, Frederic Blain, Daniel Deutsch, Craig Stewart, Chrysoula Zerva, Sheila Castilho, Alon Lavie, and George Foster. 2023. Results of WMT23 metrics shared task: Metrics might be guilty but references are not innocent. In *Proceedings of the Eighth Conference on Machine Translation*, pages 578–628, Singapore. Association for Computational Linguistics.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, Eleftherios Avramidis, Tom Kocmi, George Foster, Alon Lavie, and André F. T. Martins. 2022. Results of WMT22 metrics shared task: Stop using BLEU – neural metrics are better and more robust. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 46–68, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Leo Gao, John Schulman, and Jacob Hilton. 2023. Scaling Laws for Reward Model Overoptimization. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 10835–10866. PMLR.

Adam Gleave, Michael Dennis, Cody Wild, Neel Kant, Sergey Levine, and Stuart Russell. 2020. Adversarial policies: Attacking deep reinforcement learning. In *International Conference on Learning Representations*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783*.

Yoichiro Hasebe and Jae-Ho Lee. 2015. Introducing a readability evaluation system for Japanese language education. In *Proceedings of the 6th international conference on computer assisted systems for teaching & learning Japanese*, pages 19–22.

Sunghwan Kim, Dongjin Kang, Taeyoon Kwon, Hyungjoo Chae, Dongha Lee, and Jinyoung Yeo. 2025. Rethinking Reward Model Evaluation Through the Lens of Reward Overoptimization. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13252–13280, Vienna, Austria. Association for Computational Linguistics.

Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Marzena Karpinska, Philipp Koehn, Benjamin Marie, Christof Monz, Kenton Murray, Masaaki Nagata, Martin Popel, Maja Popović, Mariya Shmatova, Steinthór Steingrímsson, and Vilém Zouhar. 2024. Findings of the WMT24 general machine translation shared task: The LLM era is here but MT is not solved yet. In *Proceedings of the Ninth Conference on Machine Translation*, pages 1–46, Miami, Florida, USA. Association for Computational Linguistics.

Moxin Li, Yuantao Zhang, Wenjie Wang, Wentao Shi, Zhuo Liu, Fuli Feng, and Tat-Seng Chua. 2025. Self-Improvement Towards Pareto Optimality: Mitigating Preference Conflicts in Multi-Objective Alignment. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 11010–11031, Vienna, Austria. Association for Computational Linguistics.

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

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*.

Alexander Pan, Kush Bhatia, and Jacob Steinhardt. 2022. The effects of reward misspecification: Mapping and mitigating misaligned models. In *International Conference on Learning Representations*.

Rafael Rafailov, Yaswanth Chittepu, Ryan Park, Harshit Sikchi, Joey Hejna, W. Bradley Knox, Chelsea Finn, and Scott Niekum. 2024. Scaling laws for reward model overoptimization in direct alignment algorithms. In *Advances in Neural Information Processing Systems*, volume 37, pages 126207–126242. Curran Associates, Inc.

Abhinav Rastogi, Albert Q Jiang, Andy Lo, Gabrielle Berrada, Guillaume Lample, Jason Rute, Joep Barmentlo, Karmesh Yadav, Kartik Khandelwal, Khyathi Raghavi Chandu, et al. 2025. Magistral. *arXiv preprint arXiv:2506.10910*.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning Robust Metrics for Text Generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. 2024. Deepseekmath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint arXiv:2402.03300*.

Wei Shen, Rui Zheng, Wenyu Zhan, Jun Zhao, Shihan Dou, Tao Gui, Qi Zhang, and Xuan-Jing Huang. 2023. Loose lips sink ships: Mitigating Length Bias in Reinforcement Learning from Human Feedback. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2859–2873.

Prasann Singhal, Tanya Goyal, Jiacheng Xu, and Greg Durrett. 2024. A Long Way to Go: Investigating Length Correlations in RLHF. In *First Conference on Language Modeling*.

Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. 2022. Defining and Characterizing Reward Gaming. In *Advances in Neural Information Processing Systems*, volume 35, pages 9460–9471. Curran Associates, Inc.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems*, volume 33, pages 3008–3021. Curran Associates, Inc.

Mykola Trokhymovych, Indira Sen, and Martin Gerlach. 2024. An Open Multilingual System for Scoring Readability of Wikipedia. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6296–6311, Bangkok, Thailand. Association for Computational Linguistics.

Changyi Xiao, Mengdi Zhang, and Yixin Cao. 2025. BNPO: Beta Normalization Policy Optimization. *arXiv preprint arXiv:2506.02864*.

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

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

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc.

Zhanhui Zhou, Jie Liu, Jing Shao, Xiangyu Yue, Chao Yang, Wanli Ouyang, and Yu Qiao. 2024. Beyond One-Preference-Fits-All Alignment: Multi-Objective Direct Preference Optimization. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10586–10613, Bangkok, Thailand. Association for Computational Linguistics.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2020. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

A Related Work: Variants of GRPO

Several algorithms have been proposed to improve GRPO. Xiao et al. (2025) uses adaptive reward normalization with a Beta distribution to improve training stability and precision, outperforming REINFORCE and GRPO in reasoning tasks, and it dynamically normalizes rewards, enhancing policy optimization. In addition, LLMs have been developed that apply an improved method of GRPO that performs length normalization sequentially and normalizes the advantage function batch by batch (Rastogi et al., 2025). Dynamic Policy Optimization (Feng et al., 2025) is an extension of GRPO that enables large language models to perform adaptive, multi-turn reasoning in dynamic environments. In experiments, DyPO outperformed existing methods consistently in interactive decision-making and reasoning. Zheng et al. (2025) proposed the new GRPO method to use sequence likelihood for importance sampling.

Our contribution is distinct from these studies as we focus on low-resource settings where a reliable reward model is not available. MO-GRPO is orthogonal to these ideas and can be combined with these algorithms.

B Formal Formulation of GRPO

The formal formulation of GRPO is as follows:

$$J_{\text{GRPO}}(\theta) = \mathbb{E}_{q,\{o_g\}\sim\pi_{\theta_{\text{ref}}}} \Bigg[\frac{1}{G}\sum_{g=1}^{G} \frac{1}{|o_g|} \tag{8}$$

$$\min\Bigg(\frac{\pi_\theta(o_g \mid q)}{\pi_{\theta_{\text{ref}}}(o_g \mid q)} A_g, \tag{9}$$

$$\text{clip}\Big(1-\epsilon,\, 1+\epsilon,\, \frac{\pi_\theta(o_g \mid q)}{\pi_{\theta_{\text{ref}}}(o_g \mid q)}\Big) A_g \Bigg) \tag{10}$$

$$- \beta\, \text{KL}(\pi_\theta \,\|\, \pi_{\theta_{\text{ref}}}) \Bigg]. \tag{11}$$

where ϵ is a threshold parameter.

C Correlation Analysis of Dr. GRPO

In Dr. GRPO, the advantage function is defined as:

$$A_g^{\text{Dr}} = \sum_{i=1}^{K} R_i(q, o_g) \;-\; \text{mean}_{\mathbf{o}}\Big(\sum_{i=1}^{K} R_i(q, \mathbf{o})\Big). \tag{12}$$

Theorem 3 (Correlation each reward function and advantage function with Dr. GRPO)**.** *Assume that $G \to \infty$. The correlation coefficient between an individual reward function R_i and the advantage A is the ratio of the standard deviation of R_i to the standard deviation of the total reward.*

$$\text{Corr}(R_i, A_g^{Dr}) = \tag{13}$$

$$\frac{\sigma_i^2 + X}{\sqrt{\sigma_i^2 \left(\sum_{j=1}^{K} \sigma_j^2 + \sum_{j\neq l} \sum_{l\neq j} \text{Cov}\left(R_j, R_l\right) \right)}} \tag{14}$$

where $X = \sum_{j\neq i} \text{Cov}(R_i, R_j)$.

Proof. From here on, for simplicity, we omit the notation for prompt q and optional output o_g (e.g., $R_i(q, o_g) \to R_i$). We assume the number of samples $G \to \infty$, which allows the sample statistics to approximate the true population parameters. Let $R_1, \ldots, R_i$ be K reward functions. We assume they are uncorrelated, such that $\text{Cov}[R_i, R_j] = 0$ for all $i \neq j$. Let $\mu_i = \mathbb{E}[R_i]$ and $\sigma_i^2 = \text{Var}[R_i]$ denote the mean and variance of the i-th reward, respectively.

$$A^{\text{Dr}} \;:=\; \mathbf{R} - \mu, \qquad \mu = \mathbb{E}[\mathbf{R}]. \tag{15}$$

$$\text{Cov}(R_i, A^{\text{Dr}}) = \text{Cov}\left(R_i, \mathbf{R} - \mathbb{E}[\mathbf{R}]\right) \tag{16}$$

$$= \text{Cov}(R_i, \mathbf{R}) \tag{17}$$

$$= \text{Cov}\left(R_i, \sum_{j=1}^{K} R_j\right) \tag{18}$$

$$= \sum_{j=1}^{K} \text{Cov}(R_i, R_j) \tag{19}$$

$$= \left(\text{Cov}(R_i, R_i) + \underbrace{\sum_{j\neq i} \text{Cov}(R_i, R_j)}_{X} \right) \tag{20}$$

$$= (\text{Var}[R_i] + X) = \sigma_i^2 + X \tag{21}$$

The correlation coefficient between the i-th reward and the advantage is:

$$\text{Corr}(R_i, A^{\text{Dr}}) = \frac{\text{Cov}\left(R_i, A^{\text{Dr}}\right)}{\sqrt{\text{Var}\left[R_i\right] \text{Var}\left[A^{\text{Dr}}\right]}} \tag{22}$$

Finally, we can get the correlation between each reward function and advantage function with Dr.

GRPO.

$$\mathrm{Corr}(R_i, A^{\mathrm{Dr}}) = \frac{\mathrm{Cov}(R_i, A^{\mathrm{Dr}})}{\sqrt{\mathrm{Var}[R_i]\,\mathrm{Var}[A^{\mathrm{Dr}}]}} \tag{23}$$

$$= \frac{\sigma_i^2 + X}{\sqrt{\sigma_i^2 \cdot \mathrm{Var}[A^{\mathrm{Dr}}]}} \tag{24}$$

$$= \frac{\sigma_i^2 + X}{\sqrt{\sigma_i^2 \left(\sum_{j=1}^{K} \sigma_j^2 + \sum_{j \neq l} \sum_{l \neq j} \mathrm{Cov}\left(R_j, R_l\right) \right)}} \tag{25}$$

□

If there is no correlation between rewards, the following applies:

$$\mathrm{Corr}(R_i, A^{\mathrm{Dr}}) = \frac{\sigma_i}{\sigma} \tag{26}$$

In other words, Dr. GRPO implies that learning is biased toward rewards with large variances.

D Reward hacking Examples

We show the results of the outputs during training (Table 8), and the other LLMs cause the reward hacking behavior (Table 9 and Table 10). In Table 9 and Table 10, it can be seen that both Qwen and Sarashina use more non-Japanese languages in GRPO to increase the jReadabillity score (reward hacking).

E Proof of Theorem and Proposition

E.1 Proof of Theorem 1

From here on, for simplicity, we omit the notation for prompt q and optional output o_g (e.g., $R_i(q, o_g) \to R_i$). We assume the number of samples $G \to \infty$, which allows the sample statistics to approximate the true population parameters. Let $R_1, \ldots, R_i$ be K reward functions. Let $\mu_i = \mathbb{E}[R_i]$ and $\sigma_i^2 = \mathrm{Var}[R_i]$ denote the mean and variance of the i-th reward, respectively.

$$\mathrm{Cov}(R_i, A) = \mathrm{Cov}\left(R_i, \frac{\mathbf{R} - \mathbb{E}[\mathbf{R}]}{\sigma}\right) \tag{27}$$

$$= \frac{1}{\sigma}\,\mathrm{Cov}(R_i, \mathbf{R}) \tag{28}$$

$$= \frac{1}{\sigma}\,\mathrm{Cov}\left(R_i, \sum_{j=1}^{K} R_j\right) \tag{29}$$

$$= \frac{1}{\sigma} \sum_{j=1}^{K} \mathrm{Cov}(R_i, R_j) \tag{30}$$

$$= \frac{1}{\sigma}\left(\mathrm{Cov}(R_i, R_i) + \sum_{j \neq i} \mathrm{Cov}(R_i, R_j)\right) \tag{31}$$

$$= \frac{1}{\sigma}(\mathrm{Var}[R_i] + X) = \frac{\sigma_i^2 + X}{\sigma} \tag{32}$$

Finally, we can get the correlation between each reward function and advantage function with GRPO.

$$\mathrm{Corr}(R_i, A) = \frac{\mathrm{Cov}(R_i, A)}{\sqrt{\mathrm{Var}[R_i]\,\mathrm{Var}[A]}} \tag{33}$$

$$= \frac{\frac{\sigma_i^2 + X}{\sigma}}{\sqrt{\sigma_i^2 \cdot 1}} \tag{34}$$

$$= \frac{\sigma_i^2 + X}{\sigma \sigma_i} \tag{35}$$

The intuitive understanding of this proposition is that when a reward function R_i has a negative correlation with other reward functions, X becomes negative, thereby reducing the influence of R_i on the advantage function.

E.2 Proof of Theorem 2

From here on, for simplicity, we omit the notation for prompt q and optional output o_g (e.g., $R_i(q, o_g) \to R_i$). We assume the number of samples $G \to \infty$, which allows the sample statistics to approximate the true population parameters. Let $R_1, \ldots, R_i$ be K reward functions. Let $\mu_i = \mathbb{E}[R_i]$ and $\sigma_i^2 = \mathrm{Var}[R_i]$ denote the mean and variance of the i-th reward, respectively.

$$A^{\mathrm{MO}} := \sum_{j=1}^{K} \frac{R_j - \mathrm{mean}(R_j)}{\mathrm{std}(R_j)}. \tag{36}$$

Instruction	**Translate the following English into easily readable Japanese.**\nOver the past decade, our lives have changed through technology, with many working from home, ...	jReadability↑	BLEURT↑
GRPO ($\frac{1}{10}T$)	過去の10年で、技術の進化により、多くの人が家で働いており ...	0.33	0.69
GRPO ($\frac{1}{3}T$)	Over the past ten years, our lives have changed a lot because of technology. ...	**0.99**	0.57
GRPO (T)	Over the past ten years, our lives have changed a lot because of technology. ...	**0.94**	0.57
MO-GRPO (T)	過去の10年で、技術の進化により、多くの人がホームワークから仕事をしているようになり...	0.40	0.69

Table 8: (Machine translation) Generation examples of GRPO and MO-GRPO by Llama (Llama-3.2-3B-Instruct). T is the total steps. **GRPO optimizes only the Japanese readability score (jReadability) by avoiding using difficult Japanese words, eventually stops using any Japanese characters**, ignoring the translation accuracy score (BLEURT), resulting in generating non-Japanese text, which defeats the purpose of the translation. On the other hand, MO-GRPO evenly optimizes both objectives, achieving improvement on both objectives as intended.

We first calculate $\mathrm{Var}[A^{\mathrm{MO}}]$.

$$\mathrm{Var}\left(A^{\mathrm{MO}}\right) = \sum_{j=1}^{K} 1 + \sum_{l\neq j} \frac{\mathrm{Cov}\left(R_l, R_j\right)}{\sigma_l \sigma_j} \tag{37}$$

$$= K + \underbrace{\sum_{j=1}^{K}\sum_{l\neq j}^{K} \frac{\mathrm{Cov}\left(R_l, R_j\right)}{\sigma_l \sigma_j}}_{Y} \tag{38}$$

The corresponding correlation is:

$$\mathrm{Corr}(R_i, A^{\mathrm{MO}}) = \frac{\mathrm{Cov}(R_i, \sum_{j=1}^{K} \frac{R_j - \mu_j}{\sigma_j})}{\sigma_i \sqrt{K+Y}} \tag{39}$$

$$= \frac{\sum_{j=1}^{K} \frac{1}{\sigma_j} \mathrm{Cov}\left(R_i, R_j\right)}{\sigma_i \sqrt{K+Y}} \tag{40}$$

$$= \frac{\sigma_i \sum_{j=K} \frac{\mathrm{Cov}(R_i, R_j)}{\sigma_i \sigma_j}}{\sigma_i \sqrt{K+Y}} \tag{41}$$

$$= \frac{1 + \sum_{j\neq i} \frac{\mathrm{Cov}(R_i, R_j)}{\sigma_i \sigma_j}}{\sqrt{K+Y}} \tag{42}$$

$$= \frac{1+Z}{\sqrt{K+Y}} \tag{43}$$

$$\tag{44}$$

E.3 Proof of Proposition 1

For simplicity, we know μ_i and σ_i, the true mean and standard deviation of R_i over a group of outputs $\mathbf{o}$. The mean μ_i' and standard deviation σ_i' of the transformed reward R_i' are:

$$\mu_i' = \mathbb{E}[a_i R_i + b_i] = a_i \mu_i + b_i$$
$$\sigma_i' = \mathrm{std}(a_i R_i + b_i) = a_i \sigma_i \quad (\text{since } a_i > 0)$$

The i-th advantage function calculated using the transformed reward R_i' for any o is:

$$\frac{R_i'(o) - \mu_i'}{\sigma_i'} = \frac{(a_i R_i(o) + b_i) - (a_i \mu_i + b_i)}{a_i \sigma_i} \tag{45}$$

$$= \frac{a_i (R_i(o) - \mu_i)}{a_i \sigma_i} \tag{46}$$

$$= \frac{R_i(o) - \mu_i}{\sigma_i} \tag{47}$$

Since each advantage function is invariant, their sum A^{MO} is also invariant.

E.4 Proof of Proposition 2

The advantage function of GRPO's mean and standard deviation of rewards between groups remains unchanged, allowing us to focus on the value of the rewards. For simplicity, we consider two reward functions and two outputs, o_a and o_b. Assume a trade-off scenario $R_1(o_a) > R_1(o_b)$ and $R_2(o_a) < R_2(o_b)$ and $R_1(o_b) + R_2(o_b) < R_1(o_a) + R_2(o_a)$. We consider a scaling $\mathcal{R}'$ where

Method	Output	jReadability↑	BLEURT↑
Input	"People Swimming in the Swimming Pool" from 2022 is one Vicente Siso artwork that will display at Tierra del Sol Gallery beginning Jan. 13. (photo courtesy of Vicente Siso)	–	–
GRPO	"People Swimming in the Swimming Pool" 2022年はビクセン・シソオーワークスがティラードールギャラリーで1月13日から展示します。（ビクセン・シソオーフォト提供）	1.0	0.68
MO-GRPO	2022年の「泳ぎの人たち」は、1月13日からティラードールギャラリーでVICENTE SISOの作品が展示されます。（VICENTE SISOの写真提供)	0.86	0.77

Table 9: Generation examples of GRPO and MO-GRPO overoptimizing for a single reward function (readability score) at the cost of the other (translation accuracy) with Qwen. **GRPO exploits the problem of the jReadability score that it significantly increases when non-Japanese characters are used**, resulting in generating non-Japanese characters, which defeats the purpose of the translation task. On the other hand, MO-GRPO evenly optimizes both objectives, achieving improvement on both.

$R'_i = a_i R_i$ (i.e., $b_i = 0$) and derive the condition under which the preference ordering is reversed:

$$A_a > A_b \Rightarrow A'_a < A'_b.$$

where $A'_a = \frac{\sum_{i=1}^{K} R'_i(q,o_g) - \text{mean}\left(\sum_{i=1}^{K} R'_i(q,\mathbf{o})\right)}{\text{std}\left(\sum_{i=1}^{K} R'_i(q,\mathbf{o})\right)}$.

This reduces to:

$$a_1 R_1(o_b) + a_2 R_2(o_b) > a_1 R_1(o_a) + a_2 R_2(o_a) \tag{48}$$

$$\Rightarrow \frac{a_2}{a_1} > \frac{R_1(o_a) - R_1(o_b)}{R_2(o_b) - R_2(o_a)} \tag{49}$$

Such a_1 and a_2 exist, so GRPO does not hold.

F Experiment Settings in WMT

Table 11 shows the prompt to evaluate on gpt-4o-mini, and Table 12 shows the parameter settings applied in the experiment.

G Reproducibility Statement

The experiments are conducted using an NVIDIA A100 GPU with 80 GB VRAM.

All the code of the experiments will be open-sourced on publication. The datasets and models used in the experiments ar e publicly available (Table 13) except for GPT-4o-mini used for evaluation.

H Supplementary Result

We have shown cases where some reward function is learned, but we will conduct supplementary experiments to confirm whether the proposed method maintains the performance of GRPO when the reward function is not over-optimized.

I Supplementry Results of mo-reacher-v5

We show the results of GRPO (Figure 7) and MO-GRPO (Figure 8) in mo-reacher-v5.

Method	Output	jReadability↑	BLEURT↑
Input	Siso's depictions of land, water center new gallery exhibition	–	–
GRPO	「シスコが描いた土地と水の新しい展覧会がギャラリーで始まります」(This translation conveys the main idea of the English text, focusing on the subject Siso (which could be a misspelling or a pseudonym for an artist like Claude Monet, known for his water lily paintings), the action depictions, and the new exhibition at a gallery, making it suitable for elementary school children and easy to understand.)	0.77	0.61
MO-GRPO	「シスコの陸と海を描いた新しい展覧会」	0.57	0.77

Table 10: Generation examples of GRPO and MO-GRPO overoptimizing for a single reward function (readability score) at the cost of the other (translation accuracy) with Sarashina. **GRPO exploits the problem of the jReadability score that it significantly increases when non-Japanese characters are used**, resulting in generating non-Japanese characters, which defeats the purpose of the translation task. On the other hand, MO-GRPO evenly optimizes both objectives, achieving improvement on both.

J Practical Implementation

```
def MO_GRPO(reward_1,reward_2):
    combined_scores = []
    standard_deviation_reward_1  = np.std(reward_1) + 1e-6
    standard_deviation_reward_2 = np.std(reward_2) + 1e-6
    reward_1_norm = (reward_1 - np.mean(reward_1)) / standard_deviation_reward_1
    reward_2_norm = (reward_2 - np.mean(reward_2)) / standard_deviation_reward_2

    for i in range(len(group_samples)):
        combined_score = (
            (reward_1_norm[i] + reward_2_norm[i]) / np.sqrt(2)
        )
        combined_scores.append(combined_score)
```

K Training Process of GRPO and MO-GRPO with BLEURT and jReadability

We show the training process of MO-GRPO with BLEURT and jReadability by Llama and Qwen (Figure 9).

We show the training process of GRPO with BLEURT and jReadability by Llama and Qwen (Figure 10 and Figure 11).

As a neutral reviewer, please evaluate the quality of the translations and simplification task from English to Japanese provided by two AI assistants for the following users' English texts. Follow the user's instructions and select the assistant that provides the most appropriate response to user input. When evaluating, consider factors such as the accuracy of the response (similarity to the [Reference] text) and readability (easy to read for non-native Japanese speakers and a child). When starting the evaluation, compare the two responses and provide a brief explanation. Avoid bias and ensure that the order in which the responses are presented does not influence your judgment. Do not let the length of the response influence your evaluation. Do not favor the name of a specific AI assistant. Strive to be as objective as possible. After providing an explanation, output your final judgment in the following format: "[[A]]" if Assistant A is superior, "[[B]]" if Assistant B is superior, and "[[C]]" if it is a tie.

(User Input) ⟨input⟩

(Reference) ⟨reference⟩

Assistant A ⟨answer A⟩

Assistant B ⟨answer B⟩

Table 11: Prompt used for GPT-4o-mini-based evaluation.

Parameter	
temperature	0.7
learning rate	2e-6
adam beta1	0.9
adam beta2	0.99
weight decay	0.1
gradient accumulation steps	4
num generations	8
num train epochs	3
beta	0.04

Table 12: Parameter Setting of the Experiment in WMT for GRPO, Dr. GRPO, and MO-GRPO.

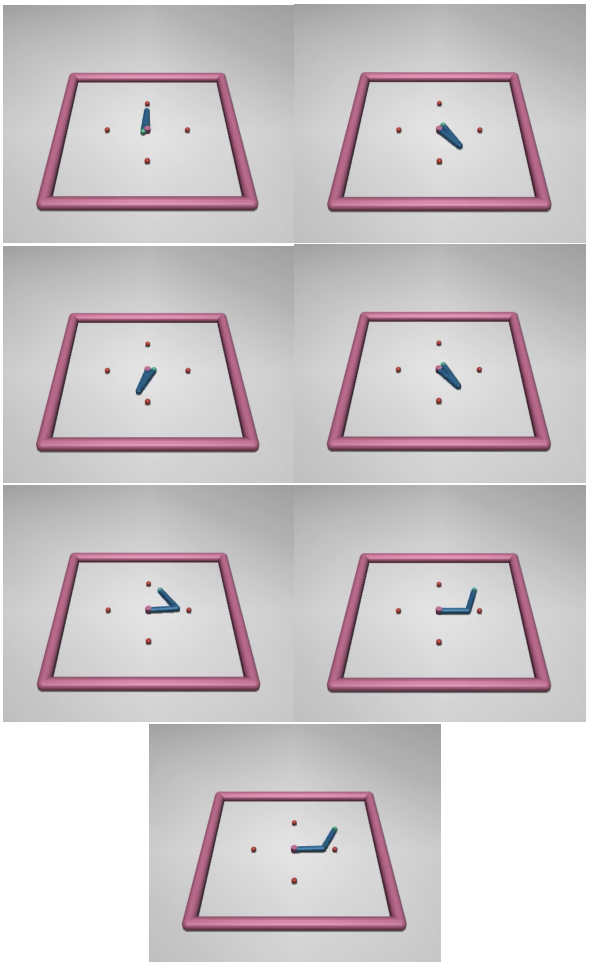

Figure 7: This shows the results of GRPO in mo-reacher-v5. GRPO's learned policy does not swing the reacher once, but rather stops in the right half, close to reward function 1 R_1.

Name	Reference
WMT (Kocmi et al., 2024)	https://github.com/wmt-conference
BLEURT (Sellam et al., 2020)	https://huggingface.co/lucadiliello/BLEURT-20
Sarashina	https://huggingface.co/sbintuitions/sarashina2.2-3b-instruct-v0.1
Qwen (Yang et al., 2025)	https://huggingface.co/Qwen/Qwen2.5-3B-Instruct
Llama (Grattafiori et al., 2024)	https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct
jReadability (Hasebe and Lee, 2015)	https://github.com/joshdavham/jreadability
Alapca (Dubois et al., 2023)	https://huggingface.co/datasets/tatsu-lab/alpaca
RM-Mistral-7B (Dong et al., 2023)	https://huggingface.co/weqweasdas/RM-Mistral-7B
TRank (Trokhymovych et al., 2024)	https://huggingface.co/trokhymovych/TRank_readability

Table 13: List of datasets and models used in the experiments.

Method	**BLEURT** ↑	**jReadability** ↑
GRPO	0.76	0.72
MO-GRPO (ours)	0.76	0.72

Table 14: Translation quality on WMT23 De-En, training dataset is WMT-21, 22.

Method	**BLEURT** ↑	**jReadability** ↑
GRPO	0.78	0.68
MO-GRPO (ours)	0.78	0.70

Table 15: Translation quality on WMT23 Ru-En, training dataset is WMT-21, 22.

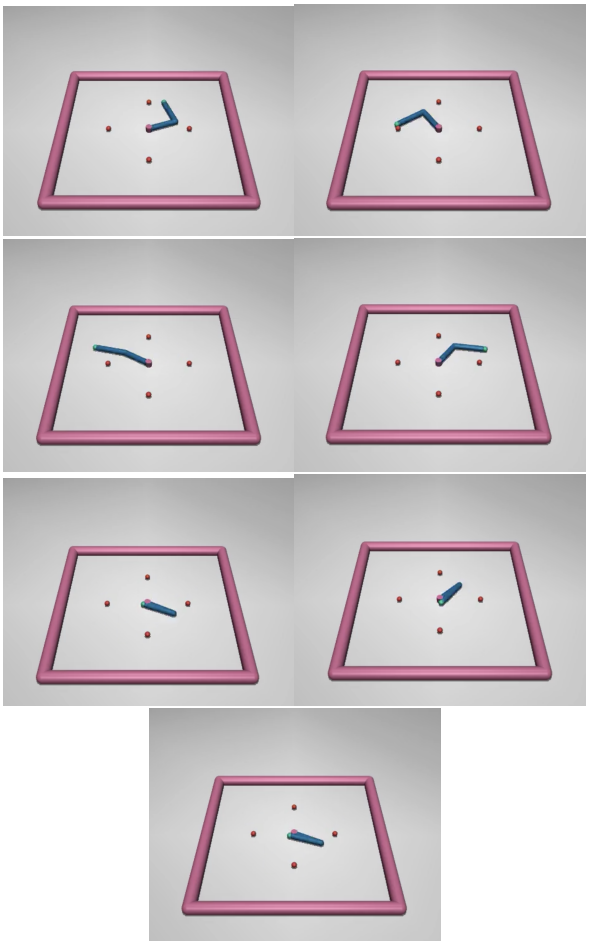

Figure 8: This shows the results of MO-GRPO in mo-reacher-v5. The policy learned by MO-GRPO successfully completes one swing of the reacher.

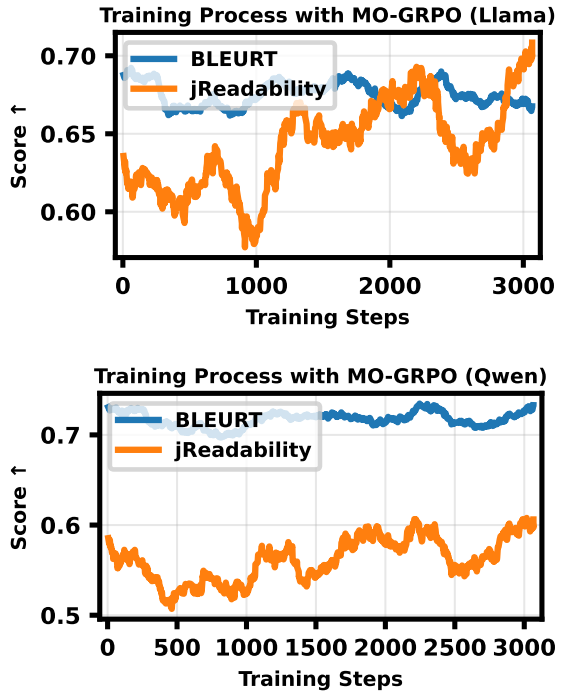

Figure 9: (Machine translation) The training process of MO-GRPO on the WMT En-Ja dataset uses BLEURT and jReadability as the reward functions by Llama, and Qwen. Unlike GRPO (Figure 5, Figure 10, and Figure 11), the figure shows MO-GRPO avoids overfitting of jReadability and prevents deterioration of BLEURT in all language models.

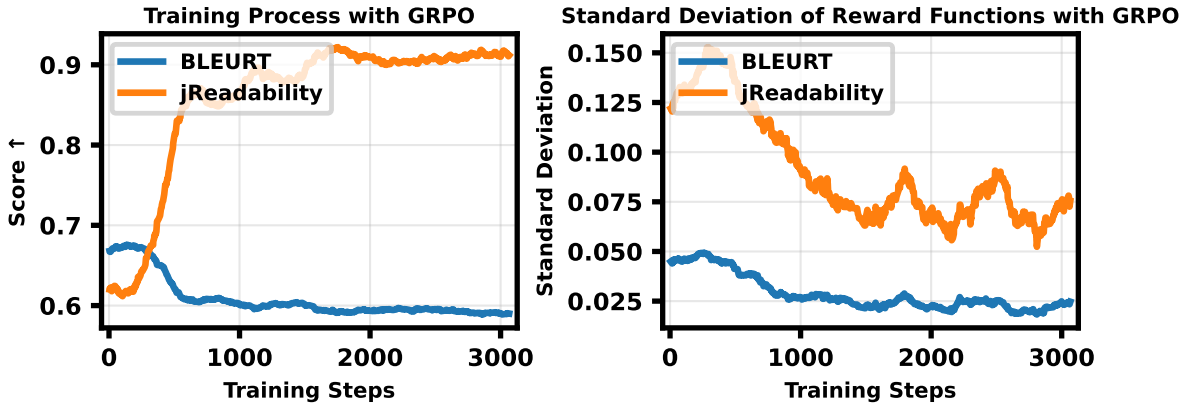

Figure 10: The training process of GRPO on the WMT En-Ja dataset uses BLEURT and jReadability as the reward functions by Llama. As the results show, GRPO overfits jReadability at the expense of BLEURT performance. As the results show, the standard deviation of jReadability is always more than BLEURT.

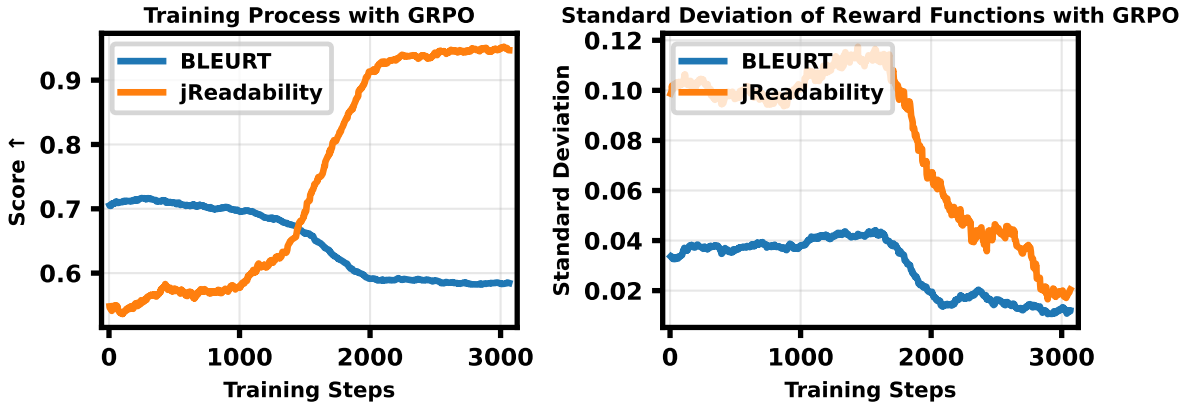

Figure 11: The training process of GRPO on the WMT En-Ja dataset uses BLEURT and jReadability as the reward functions by Qwen. As the results show, GRPO overfits jReadability at the expense of BLEURT performance. As the results show, the standard deviation of jReadability is always more than BLEURT.