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 three domains: (i) the multi-armed bandits problem, (ii) machine translation tasks on the WMT benchmark (En-Ja, En-Zh), and (iii) the 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. 2025a) with high-accuracy reward models; however, it is difficult to obtain them 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.

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 original 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, machine translation (Kocmi et al., 2024) problems, and instruction following task with multiple objectives. The result

Transactions of the Association for Computational Linguistics, vol. 14, pp. 1867–1882, 2026. https://doi.org/10.1162/TACL.a.782
Action Editor: Eric Yuan. Submission batch: 9/2025; Revision batch: 2/2026; Published 8/2026.

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.

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 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. $p_{\mathcal{Q}}$ is the distribution over the initial state (prompt) ($q \sim p_{\mathcal{Q}}$). 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 old policy $\pi_{\theta_{\text{old}}}$ and then optimizes the policy model by maximizing the following objective:

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

where β is a hyperparameter, and KL is Kullback–Leibler (KL) divergence.

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

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

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.*

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

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

The proof is in Appendix D.1.

Remark (Finite-G implementation). Theorem 1 is stated for population quantities, i.e., the advantage is defined using the population mean and standard deviation of the *total reward*. In practice, GRPO computes the advantage using group-wise (empirical) normalization over a finite number of samples G: Accordingly, the population variances/covariances in Theorem 1 are

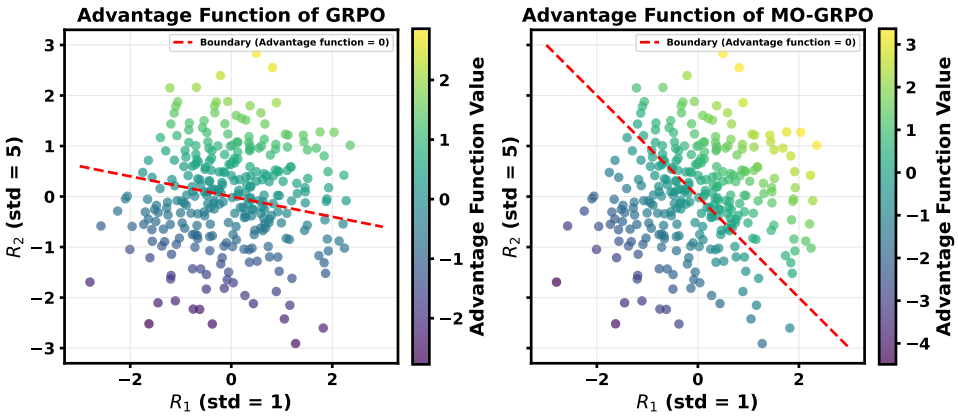


Figure 1: (Simulated experiment) Comparison of the advantage values of GRPO and MO-GRPO on a toy example with two reward functions with different sizes of variances (1 and 5). The advantage values of GRPO (left figure) are dominated by the high variation reward (R_2), indicating that the algorithm is 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.

replaced by their empirical counterparts computed from the same group:

$$\hat{\sigma}_i^2 = \widehat{\mathrm{Var}}(R_i), \qquad \hat{\sigma}^2 = \widehat{\mathrm{Var}}(R_{\mathrm{tot}}), \tag{4}$$

$$\hat{X}_i = \sum_{j \neq i} \widehat{\mathrm{Cov}}(R_i, R_j). \tag{5}$$

where $R_{\mathrm{tot}}(q, o) = \sum_{i=1}^{K} R_i(q, o)$. Therefore, for finite G, the correlation becomes a random (group-dependent) plug-in estimate $\widehat{\mathrm{Corr}}(R_i, \hat{A})$ rather than a constant. As $G \to \infty$, $(\hat{\sigma}_i^2, \hat{\sigma}^2, \hat{X}_i)$ converge to $(\sigma_i^2, \sigma^2, X_i)$, and the empirical correlation concentrates around the population expression in Theorem 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 4.

3 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.

$$\mathcal{J}(\pi_\theta) = \mathbb{E}\left[\sum_{g=1}^{G} \frac{1}{G}\frac{1}{|o_g|}\frac{\pi_\theta(o_g \mid q)}{\pi_{\theta_{\mathrm{old}}}(o_g \mid q)} A_g^{\mathrm{MO}} - \beta \mathrm{KL}(\pi_\theta, \pi_{\theta_{\mathrm{ref}}})\right], \tag{6}$$

where A_g^{MO} is defined as follows:

$$A_g^{\mathrm{MO}} = \sum_{i=1}^{K} \frac{R_i(q, o_g) - \mathrm{mean}_{\mathbf{o}}\big(R_i(q, \mathbf{o})\big)}{\mathrm{std}_{\mathbf{o}}\big(R_i(q, \mathbf{o})\big)}. \tag{7}$$

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.*

$$\mathrm{Corr}(R_i(q, o_g), A_g^{\mathrm{MO}}) = \frac{1 + Z_i}{\sqrt{K + Y}} \tag{8}$$

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

The proof is in Appendix D.2. Theorem 2 assumes $G \to \infty$, however, MO-GRPO still improves and avoids reward hacking with a small sample size in experiments (Table 6).

Remark (Finite-G implementation). Theorem 2 is stated for the true-normalized advantage, where each reward is centered and scaled by the true mean and standard deviation (μ_i, σ_i). In practice, MO-GRPO computes the advantage using group-wise (empirical) normalization over a finite number of samples G: Accordingly, the quantities Y and Z in Theorem 2 are also replaced by their empirical counterparts computed from the same group:

$$\hat{Y} = \sum_{j=1}^{K} \sum_{l \neq j} \frac{\widehat{\mathrm{Cov}}(R_l, R_j)}{\hat{\sigma}_l \hat{\sigma}_j}, \tag{9}$$

$$\hat{Z}_i = \sum_{j \neq i} \frac{\widehat{\mathrm{Cov}}(R_i, R_j)}{\hat{\sigma}_i \hat{\sigma}_j}, \tag{10}$$

where $\widehat{\mathrm{Cov}}(\cdot, \cdot)$ denotes the sample covariance over the G candidates. Therefore, for finite G, the correlation becomes a random (group-dependent) plug-in estimate $\widehat{\mathrm{Corr}}(R_i, \hat{A}^{\mathrm{MO}})$ rather than a constant. As $G \to \infty$, $(\hat{\sigma}_i, \widehat{\mathrm{Cov}})$ converge to (σ_i, Cov), and the empirical correlation concentrates around the population expression in Theorem 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 of the advantage functions A_g^{MO} with each reward function R_i for any o_g remains constant.*

$$\mathrm{Corr}(R_i(q, o_g), A_g^{\mathrm{MO}}) = \frac{1}{\sqrt{K}} \tag{11}$$

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 4.

Simulated experiment. Figure 1 shows the comparison of the advantage functions of GRPO (Eq. 2) and MO-GRPO (Eq. 7) 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.

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:*

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

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

The proof of Proposition 1 is in Appendix D.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.*

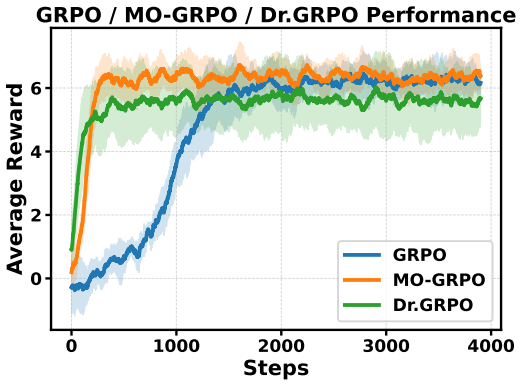


Figure 2: (Multi-armed bandit) This figure illustrates the average rewards obtained by the sum of the three reward functions: GRPO, MO-GRPO, and Dr. GRPO. As the figure shows, MO-GRPO finds a better policy faster than GRPO and Dr. GRPO.

The proof is in Appendix D.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.

4 Experiment

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

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

[1]The simplest implementation example of MO-GRPO is shown in Appendix H.

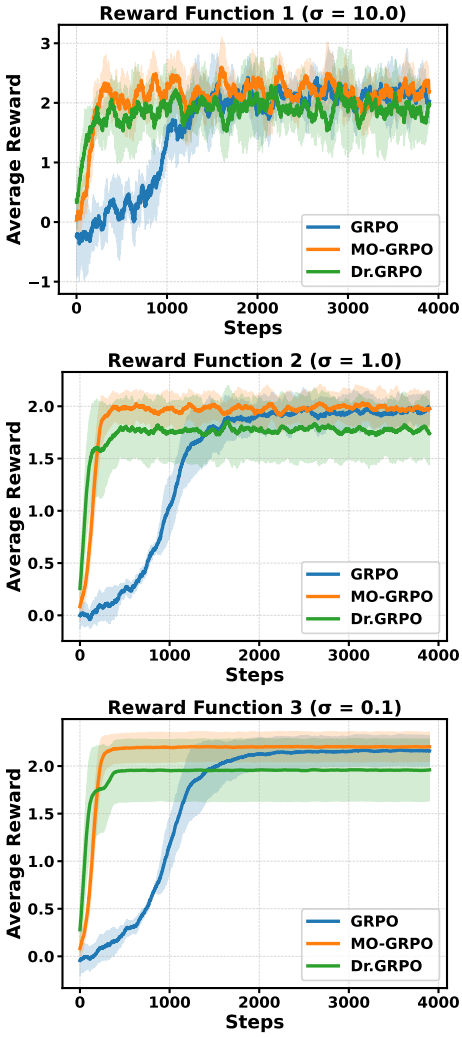


Figure 3: (Multi-arm bandit) Comparison of the three reward functions with varying variances (10, 1, and 0.1) obtained by GRPO, Dr. GRPO, and MO-GRPO. While GRPO and Dr. GRPO fail or are slow to learn the reward functions with lower variances (R_2 and R_3), MO-GRPO successfully optimizes all three reward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

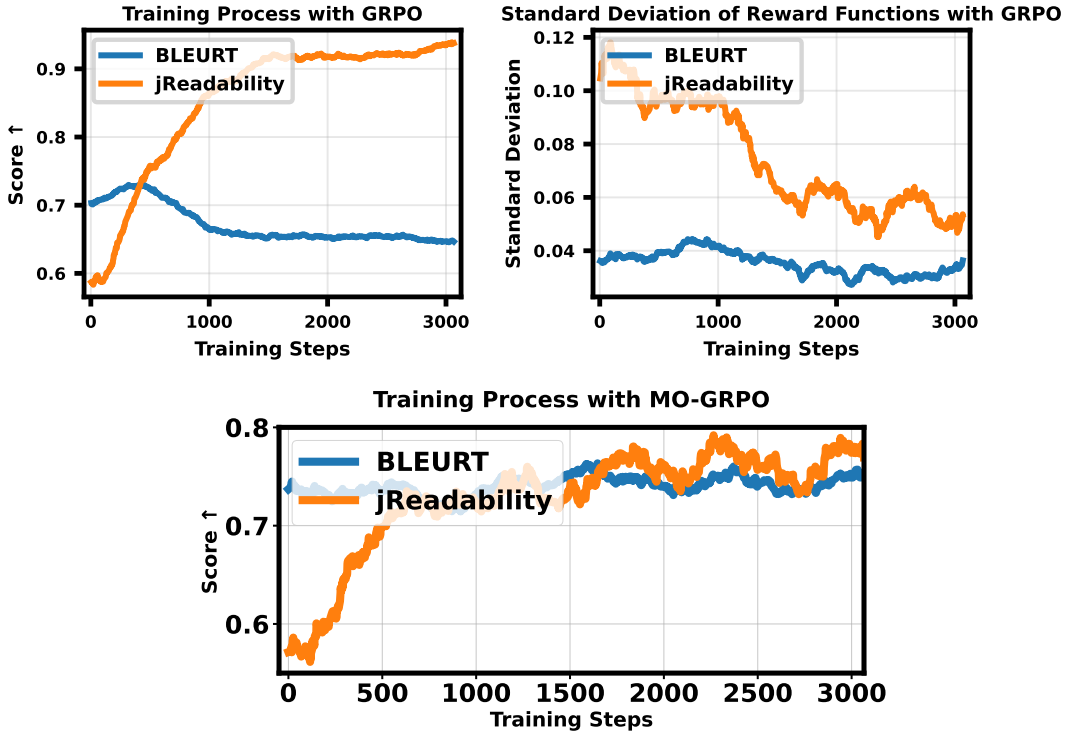


Figure 4: (Machine translation) Training curves on WMT En–Ja with BLEURT and jReadability rewards (Sarashina; sarashina2.2-3b-instruct-v0.1). **Top:** GRPO overfits jReadability at the expense of BLEURT; the standard deviation of jReadability is consistently larger than that of BLEURT. **Bottom:** MO-GRPO mitigates this issue, avoiding overfitting to jReadability and preventing BLEURT deterioration

ward 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 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.

4.2 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., 2025a), and Llama (`Llama-3.2-3B-Instruct`) (Grattafiori et al., 2024) as the base models. For the reward

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%**
Base Model	**Method**	**WMT24 (En–Zh) 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 2: (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.

(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 2 shows that, compared to the base model score, GRPO achieved a high jReadability score 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.[2]

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, the training process of MO-GRPO with Sarashina is shown in Figure 4 (Bottom), which suggests MO-GRPO avoids overfitting of jReadability and prevents degradation of BLEURT, unlike GRPO (Figure 4 (Top)).

Next, we examine the details of the results from other language models such as Qwen and Llama. The relatively limited improvement of MO-GRPO

[2]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.

in Qwen's experiments is likely attributable to Qwen not being a language model specialized for Japanese. However, Qwen outputs 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 outputting English text instead of a Japanese translation, thereby improving the jReadability. This phenomenon again did not occur with MO-GRPO.

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 2 shows the true TRank score, i.e., the score obtained without applying the -1 multiplication during evaluation.).

As shown in Table 2, MO-GRPO also appropriately treats the two reward functions across Qwen and Llama, similar to the En-Ja task. Therefore, 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 3 further quantifies this phenomenon using langdetect, a tool to predict the language of the given text.[3] We find that GRPO and Dr. GRPO frequently produce non-Chinese generations, exploiting TRank, resulting in significantly 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 3 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 4 shows the TRank with penalties under the same experimental settings as Table 2. 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).

[3] 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 3: 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 4: (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.

4.3 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 miti-

Base Model	Method	AlpacaFarm	
		RM-Mistral-7B ↑	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 5: (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.

gates the length bias problem (Shen et al., 2023; Singhal et al., 2024). Length reward function is defined as follows:

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

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.

Table 5 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.

4.4 Supplementary Results

Effect of group size. To quantify the sensitivity of the performance of MO-GRPO, we vary the number of samples $G \in \{2, 4, 6\}$ for GRPO and MO-GRPO in Table 6 on En-Ja task. As shown in Table 6, GRPO exhibits a trade-off as G increases: jReadability improves (0.77→0.87) while BLEURT degrades (0.67→0.64), consistent with over-optimization to the readability reward. MO-GRPO avoids this reward hacking: it preserves BLEURT (0.68→0.69) and keeps jReadability stable (0.75→0.76) across G, showing that MO-GRPO is robust to the choice of $G \in \{2, 4, 6\}$.

Method	#Samples	BLEURT↑	jReadability↑
Base Model (Sarashina)	–	0.66	0.70
GRPO	2	0.67	0.77
MO-GRPO	2	0.68	0.75
GRPO	4	0.65	0.83
MO-GRPO	4	0.69	0.75
GRPO	6	0.64	0.87
MO-GRPO	6	0.69	0.76

Table 6: Performance of GRPO and MO-GRPO with varying numbers of samples on BLEURT and jReadability.

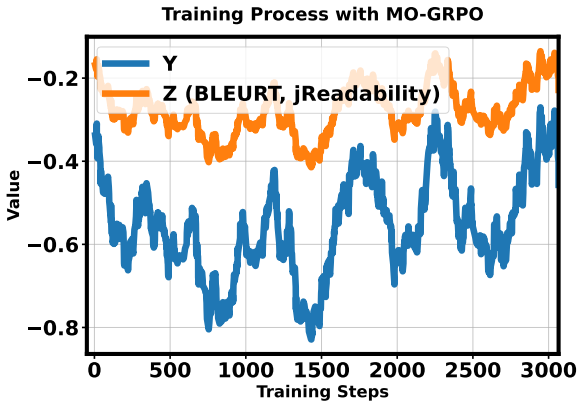
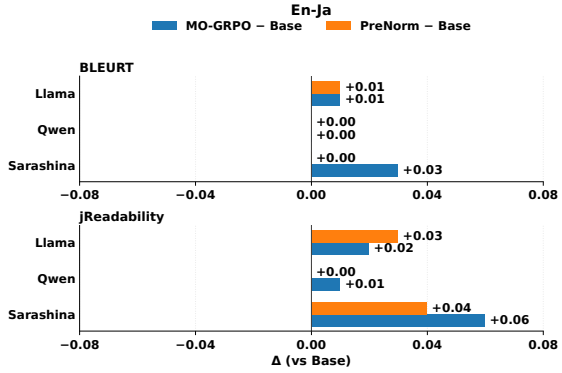


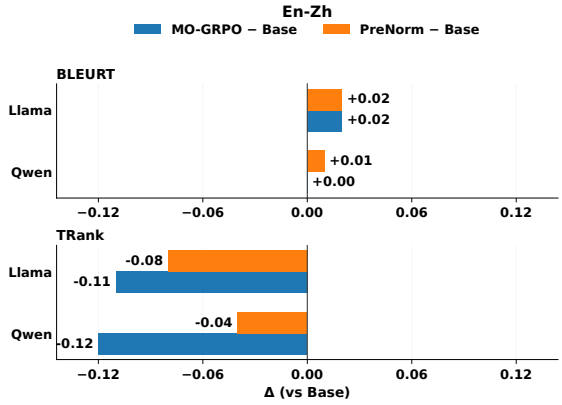
Figure 5: Empirical correlation statistics during MO-GRPO training. We show that in the two-objective case (K=2), $Z_{\text{BLEURT}} = Z_{\text{JREAD}} = \frac{Y}{2}$ by definition, explaining why Y and Z_i co-move with the same (negative) sign and an approximately 2× scale difference.

Empirical reward correlations. Corollary 1 assumes uncorrelated rewards, but BLEURT and jReadability are correlated in practice. We therefore track $Z_{\text{BLEURT}} = Z_{\text{JREAD}} = \text{Corr}(R_{\text{BLEURT}}, R_{\text{JREAD}})$ during MO-GRPO training on En–Ja (Fig. 5). For K=2, Theorem 2 yields $Z_{\text{BLEURT}} = Z_{\text{JREAD}} = Y/2$, so the figure mainly reflects the change of Z. Importantly, Z remains far from -1 (and weakens over training), consistent with stable updates and reduced over-optimization to jReadability compared to GRPO.

Effect of reward scale calibration. We estimate the reward mean and variance on WMT2021 and fix them to compute advantages (Pre-Norm). With this calibration, MO-GRPO is slightly better, and neither method shows reward hacking (Fig. 6).



(a) En-Ja dataset



(b) En-Zh dataset

Figure 6: Comparison between MO-GRPO and Pre-Norm with fixed mean and standard deviation at WMT 2024. Neither exhibits reward hacking behavior, though MO-GRPO shows a slight performance improvement.

This suggests that the key is a reasonable reward scale rather than per-group normalization. Pre-Norm, however, requires an offline pass, and other WMT data (e.g., WMT2024) have different statistics among domains (Appendix F), which leads to unstable learning.

5 Related Works

Multi-objective reinforcement learning. Multi-objective Reinforcement Learning studies sequential decision making with vector-valued rewards, where agents must trade off multiple objectives (Roijers et al., 2013). A common perspective seeks to approximate a set of Pareto-optimal solutions to cover multiple trade-offs (Roijers et al., 2013; Hayes et al., 2022). Representative approaches explicitly maintain sets of non-dominated policies during learning (Moffaert and Nowé, 2014).

Reward normalization in RL. van Hasselt et al. (2016) normalizes value targets online to support learning across many orders of magnitude, while return-based scaling provides a simple normalization trick for improved robustness (Schaul et al., 2021). Recent analyses show that PPO is also sensitive to reward scale and that DreamerV3-inspired implementation tricks can improve reward-scale robustness (Sullivan et al., 2023).

Recent variants of GRPO and relation to MO-GRPO. DAPO extends GRPO with asymmetric clipping, difficulty-adaptive sampling, token-level training, and explicit length control (Yu et al., 2025). GSPO moves from token-level to sequence-level importance ratios with sequence-level clipping (Zheng et al., 2025), SSPO uses subsentence-level ratios to bridge token and sequence-level updates (Yang et al., 2025b), and collapse mitigation via entropy control (Simoni et al., 2025), and improved compute efficiency via replay or pruning (Li et al., 2025; Lin et al., 2025). In contrast, MO-GRPO focuses on the objective problem, combining and normalizing multiple objectives to avoid imbalance scaling. MO-GRPO is also applicable in many of the related studies described above.

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 three domains: (i) the multi-armed bandits problem, (ii) machine translation tasks on the WMT benchmark (En-Ja, En-Zh), and (iii) the instruction following task. MO-GRPO consistently avoids reward hacking and shows improvements in

task-specific metrics (e.g., BLEURT, jReadability) and learning stability.

Acknowledgments

We sincerely thank the Action Editor, Eric Yuan, and the anonymous reviewers for their insightful comments and suggestions.

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.

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.

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.

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:2407.21783v3*.

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.

Hado P van Hasselt, Arthur Guez, Arthur Guez, Matteo Hessel, Volodymyr Mnih, and David Silver. 2016. Learning values across many orders of magnitude. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Conor F. Hayes, Roxana Radulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa Zintgraf, Richard Dazeley, Fredrik Heintz, Enda Howley, Athirai A. Irissappane, Patrick Mannion, Ann Nowe, Gabriel De Oliveira Ramos, Marcello Restelli, Peter

Vamplew, and Diederik M. Roijers. 2022. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*, 36(1):26.

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.

Siheng Li, Zhanhui Zhou, Wai Lam, Chao Yang, and Chaochao Lu. 2025. RePO: Replay-Enhanced Policy Optimization. *arXiv preprint arXiv:2506.09340*.

Zhihang Lin, Mingbao Lin, Yuan Xie, and Rongrong Ji. 2025. CPPO: Accelerating the Training of Group Relative Policy Optimization-Based Reasoning Models. *arXiv preprint arXiv:2503.22342*.

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:2412.19437v2*.

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:2503.20783v1*.

Kristof Van Moffaert and Ann Nowé. 2014. Multi-Objective Reinforcement Learning using Sets of Pareto Dominating Policies. *Journal of Machine Learning Research*, 15(107):3663–3692.

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.

Diederik M Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. 2013. A Survey of Multi-Objective Sequential Decision-Making. *Journal of Artificial Intelligence Research*, 48:67–113.

Tom Schaul, Georg Ostrovski, Iurii Kemaev, and Diana Borsa. 2021. Return-based Scaling: Yet Another Normalisation Trick for Deep RL. *arXiv preprint arXiv:2105.05347*.

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:2402.03300v3*.

Wei Shen, Rui Zheng, Wenyu Zhan, Jun Zhao, Shihan Dou, Tao Gui, Qi Zhang, and Xuanjing 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, Singapore. Association for Computational Linguistics.

Marco Simoni, Aleksandar Fontana, Giulio Rossolini, Andrea Saracino, and Paolo Mori. 2025. GTPO: Stabilizing Group Relative Policy Optimization via Gradient and Entropy Control. *arXiv preprint arXiv:2508.03772*.

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.

Ryan Sullivan, Akarsh Kumar, Shengyi Huang, John Dickerson, and Joseph Suarez. 2023. Reward Scale Robustness for Proximal Policy Optimization via DreamerV3 Tricks. In *Advances in Neural Information Processing Systems*, volume 36, pages 1352–1362. 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.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025a. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388v1*.

Kun Yang, Yanmeng Wang, Zhigen Li, et al. 2025b. SSPO: Subsentence-level Policy Optimization. *arXiv preprint arXiv:2511.04256*.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. 2025. DAPO: An Open-Source LLM Reinforcement Learning System at Scale. *arXiv preprint arXiv:2503.14476*.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. 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.

A Limitations

There are several limitations in this paper. (1) Our theoretical guarantees rely on asymptotic conditions, such as the number of samples approaching infinity. In practice, however, policies are updated using finite samples. (2) Our goal is to align with human preferences, but we relied on automatic evaluators instead of conducting human evaluations. (3) We did not validate MO-GRPO in more complex tasks, such as three or more objective problems in the real world. (4) MO-GRPO currently combines objectives with equal coefficients after per-objective normalization. Equal weighting is reasonable when objectives are intended to be equally important, and the main failure mode is scale mismatch. It can be problematic when objectives reflect asymmetric priorities (e.g., safety vs. style) or when reward models differ in reliability. A natural extension is weighted MO-GRPO, $A_g = \sum_i w_i \tilde{A}_{g,i}$, where w_i encodes preferences or confidence; we leave systematic weighting strategies to future work.

B 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 upon publication. The datasets and models used in the experiments are publicly available (Table 7) except for GPT-4o-mini used for evaluation.

C Correlation Analysis of Dr. GRPO

In Dr. GRPO, the advantage function is defined as $A_g^{\text{Dr}} = \sum_{i=1}^{K} R_i(q, o_g) - \text{mean}_{\mathbf{o}}\left(\sum_{i=1}^{K} R_i(q, \mathbf{o})\right)$.

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.*

$$\text{Corr}(R_i, A_g^{Dr}) = \frac{\sigma_i^2 + X_i}{\sqrt{\sigma_i^2 \left(\sum_{j=1}^{K} \sigma_j^2 + \sum_{j \neq l} \sum_{l \neq j} \text{Cov}\left(R_j, R_l\right) \right)}}$$

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

D Proof of Theorem and Proposition

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.

D.1 Proof of Theorem 1

Let $R_1, \ldots, R_K$ be K reward functions. 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.

$$\begin{aligned}
&\text{Cov}(R_i, A) = \text{Cov}\left(R_i, \frac{\mathbf{R} - \mathbb{E}[\mathbf{R}]}{\sigma}\right) \\
&= \frac{1}{\sigma}\left(\text{Cov}(R_i, R_i) + \sum_{j \neq i} \text{Cov}(R_i, R_j)\right) \\
&= \frac{1}{\sigma}(\text{Var}[R_i] + X_i) = \frac{\sigma_i^2 + X_i}{\sigma}
\end{aligned}$$

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

$$\begin{aligned}
\text{Corr}(R_i, A) &= \frac{\text{Cov}(R_i, A)}{\sqrt{\text{Var}[R_i]\,\text{Var}[A]}} \\
&= \frac{\sigma_i^2 + X_i}{\sigma \sigma_i}
\end{aligned}$$

D.2 Proof of Theorem 2

Let $R_1, \ldots, R_K$ be K reward functions. 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.

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

$$\begin{aligned}
\text{Var}\left(A^{\text{MO}}\right) &= \sum_{j=1}^{K} 1 + \sum_{l \neq j} \frac{\text{Cov}\left(R_l, R_j\right)}{\sigma_l \sigma_j} \\
&= K + \underbrace{\sum_{j=1}^{K} \sum_{l \neq j}^{K} \frac{\text{Cov}\left(R_l, R_j\right)}{\sigma_l \sigma_j}}_{Y}
\end{aligned}$$

The corresponding correlation is:

$$\begin{aligned}
\text{Corr}(R_i, A^{\text{MO}}) &= \frac{\text{Cov}(R_i, \sum_{j=1}^{K} \frac{R_j - \mu_j}{\sigma_j})}{\sigma_i \sqrt{K + Y}} \\
&= \frac{1 + Z_i}{\sqrt{K + Y}}
\end{aligned}$$

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., 2025a)	https://huggingface.co/Qwen/Qwen2.5-3B-Instruct
Llama (Grattafiori et al., 2024)	https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct
Qwen2.5-7B-Instruct (Yang et al., 2025a)	https://huggingface.co/Qwen/Qwen2.5-7B-Instruct
Llama-3-8B-Instruct (Grattafiori et al., 2024)	https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct
jReadability (Hasebe and Lee, 2015)	https://github.com/joshdavham/jreadability
Alpaca (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 7: List of datasets and models used in the experiments.

D.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:

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

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

D.4 Proof of Proposition 2

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 $R_i' = a_i R_i$ (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 $A_a' = \frac{\sum_{i=1}^K R_i'(q,o_g) - \mathrm{mean}\left(\sum_{i=1}^K R_i'(q,\mathbf{o})\right)}{\mathrm{std}\left(\sum_{i=1}^K R_i'(q,\mathbf{o})\right)}$.

This reduces to:

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

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

E Large-Scale Models

To verify that MO-GRPO is not specific to smaller language models, we replicate the same En-Ja training and evaluation on two larger instruction-tuned models, Qwen2.5-7B-Instruct (Qwen2.5-7B) and Llama3-8B-Instruct (Llama3-8B). We compare GRPO and MO-GRPO, and

show BLEURT and jReadability (Table 8). On Qwen2.5-7B, GRPO improves jReadability without degrading BLEURT, indicating no clear reward-hacking trade-off in this setting; MO-GRPO yields a small BLEURT gain while keeping readability competitive. In contrast, on Llama3-8B, GRPO increases jReadability but decreases BLEURT, suggesting a slight tendency to over-optimize jReadability at the expense of BLEURT. MO-GRPO mitigates this behavior by improving BLEURT while maintaining jReadability, demonstrating that multi-objective balancing remains important even for large-scale models.

		En-Ja	
Base Model	**Method**	**BLEURT↑**	**jReadability↑**
Qwen2.5-7B	Base Model	0.68	0.66
	GRPO	0.68	0.69
	MO-GRPO (ours)	0.69	0.67
Llama3-8B	Base Model	0.59	0.89
	GRPO	0.57	0.92
	MO-GRPO (ours)	0.62	0.89

Table 8: **Large-scale model results on En–Ja.** Two larger instruction-tuned LMs and compare Base Model, GRPO, and MO-GRPO. Qwen2.5-7B Instruct exhibits no clear reward-hacking trade-off, whereas Llama3-8B Instruct shows a stronger trade-off under GRPO. MO-GRPO mitigates this by recovering BLEURT while maintaining competitive readability.

F Mean and Std score per Domain in WMT-24

We show per-domain score distributions on the WMT-24 datasets (Tables 9 and 10). Specifically, for each model and domain, we summarize the metric values with their mean and standard deviation. WMT-24 datasets exhibit larger domain-dependent variation about means and stds. As a result, since mean and std vary depending on the domain, using PreNorm may cause unstable learning when training on samples from domains with different averages.

G Experiment Settings in WMT

We show the parameter settings of the experiments in Table 11.

H Python Implementation

We show the TRL style's implementation recipe for MO-GRPO.

Metric	Model	Literary		News		Social		Speech	
		Mean	Std	Mean	Std	Mean	Std	Mean	Std
BLEURT	Llama	0.617	0.065	0.674	0.064	0.663	0.093	0.626	0.047
	Qwen	0.636	0.071	0.672	0.054	0.684	0.081	0.625	0.048
	Sarashina	0.633	0.078	0.667	0.072	0.682	0.082	0.618	0.055
jReadability	Llama	0.691	0.204	0.428	0.205	0.737	0.225	0.655	0.161
	Qwen	0.678	0.180	0.441	0.157	0.705	0.212	0.661	0.132
	Sarashina	0.741	0.174	0.504	0.215	0.741	0.211	0.706	0.159

Table 9: En-Ja (WMT-24): Per-domain mean and standard deviation of BLEURT and jReadability (Sample size n = Literary: 206, News: 149, Social: 531, Speech: 111).

Metric	Model	Literary		News		Social		Speech	
		Mean	Std	Mean	Std	Mean	Std	Mean	Std
BLEURT	Llama	0.661	0.076	0.719	0.069	0.698	0.096	0.658	0.063
	Qwen	0.712	0.077	0.760	0.059	0.747	0.085	0.695	0.061
TRank	Llama	-2.946	1.031	-0.661	1.006	-2.651	1.047	-2.855	1.532
	Qwen	-3.267	0.965	-0.865	0.885	-2.927	1.019	-3.310	1.405

Table 10: En-Zh (WMT-24): Per-domain mean and standard deviation of BLEURT and TRank (Sample size n = Literary: 206, News: 149, Social: 531, Speech: 111).

```
def MO_GRPO(rewards_per_func,
  num_generations, eps=1e-4):
    K = rewards_per_func.shape[-1]
    grouped = rewards_per_func.view(-1,
  num_generations, K)

    mean_k = nanmean(grouped, dim=1,
  keepdim=True)
    std_k  = nanstd(grouped,  dim=1,
  keepdim=True)

    # 1) per-reward advantages (MO-GRPO):
  [B, G, K]
    A_k = (grouped - mean_k) / (std_k +
  eps)
    # 2) sum across rewards -> final
  advantage: [B, G]
    A = nansum(A_k, dim=2)

    return A.view(-1)
```

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
LoRA rank	128
LoRA alpha	128

Table 11: Parameter Setting of the Experiment in WMT for GRPO, Dr. GRPO, and MO-GRPO.