RMO: Towards Better LLM Alignment via Reshaping Reward Margin Distributions

Yanchi Ru[1*], **Yue Huang**[2*], **Xiangliang Zhang**[2†]
[1]Xi'an Jiaotong University
[2]University of Notre Dame
2196123580@stu.xjtu.edu.cn, yhuang37@nd.edu, xzhang33@nd.edu

Abstract

Large Language Models (LLMs) have achieved remarkable success in instruction-following and dialogue tasks, yet aligning them with human preferences remains a critical challenge. Recent advances such as Direct Preference Optimization (DPO) simplify the alignment pipeline by bypassing explicit reward modeling, but they often suffer from suboptimal reward margin distributions, leading to weak supervision signals and reduced discriminative capacity. In this work, we propose **Reward Margin Optimization (RMO)**, a framework that reshapes reward margin distributions during training to improve alignment performance. RMO comprises three components: (1) a Dual Denoising Filtering strategy that filters ambiguous and noisy preference pairs based on reward margin dynamics; (2) Batch Margin Diversification, which maximizes intra-batch margin variance to enhance learning signal diversity; and (3) Pairwise Margin Amplification, an auxiliary regularization term that encourages larger margins between preferred and dispreferred responses. Extensive experiments on multiple LLMs and datasets demonstrate that RMO consistently improves win rates over strong baselines such as DPO and SimPO, while remaining compatible with various preference-based optimization methods. Our results highlight the critical role of reward margin distribution in preference alignment and establish RMO as an effective and scalable enhancement to existing alignment techniques.

Code — https://github.com/ryc2001/RMO

Introduction

Large Language Models (LLMs) have demonstrated impressive capabilities across a variety of real-world applications (Hadi et al. 2023; Zhou et al. 2025). To make LLMs better reflect human values and intent, Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al. 2022) has become a cornerstone technique for aligning language models with human preferences. Despite the effectiveness of RLHF, traditional methods like PPO (Schulman et al. 2017) often suffer from high complexity and instability during optimization.

Recently, Direct Preference Optimization (DPO) (Rafailov et al. 2023) and related methods (Meng, Xia, and Chen 2024; Ethayarajh et al. 2024) have gained traction by leveraging direct preference comparisons instead of explicit reward modeling, simplifying the alignment pipeline while achieving competitive results. However, the reward margin distributions, which capture the differences between the scores of preferred and non-preferred responses, are often suboptimal and may diminish the overall alignment quality. Specifically, there are three main challenges: 1) **Ambiguous and Noisy Training Signals**: Many sample pairs exhibit small or ambiguous reward margins, often due to inherent noise or inconsistencies in the training data (Wu et al. 2024). These noisy pairs can obscure the true preference signal and potentially mislead the alignment process, reducing the effectiveness of learning; 2) **Skewed Margin Distribution**: In practice, the reward margin distribution is often imbalanced, with a majority of training batches failing to achieve a well-balanced spread of margin values, resulting in weak and unstable training signals for some batches and impeding effective optimization. As demonstrated in Figure 1 and Figure 2, we present the performance of two batch splits for the same dataset. Training with high-variance reward margin batches (i.e., more balanced margin distribution in each batch) largely stabilizes the loss curve, which in turn enhances the alignment effect of the model. 3) **Insufficient Separation**: The model's optimization objective does not explicitly encourage a larger separation between positive and negative samples, limiting its ability to robustly capture preference distinctions. The recent study (Razin et al. 2025) also highlighted that a good reward model should exhibit higher variance in the RLHF objective landscape, rather than merely achieving high accuracy. In other words, the LLM should demonstrate stronger discriminative power between positive and negative sample pairs, resulting in larger reward margins.

To address these practical limitations in preference-based alignment, we introduce **Reward Margin Optimization (RMO)**, a method that systematically reshapes reward margin distributions throughout the training process. The central motivation of RMO is that effective preference alignment requires not only a clear separation between positive and negative examples but also learning signals that are well-

[*]These authors contributed equally.
[†]Xiangliang Zhang is the corresponding author.

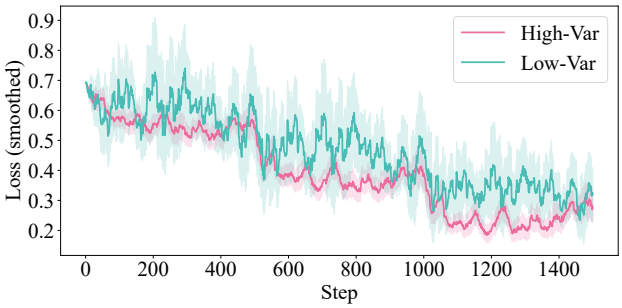


Figure 1: Loss comparison between different split strategies for Llama-3.2-1B-Instruct on Anthropic HH dataset.

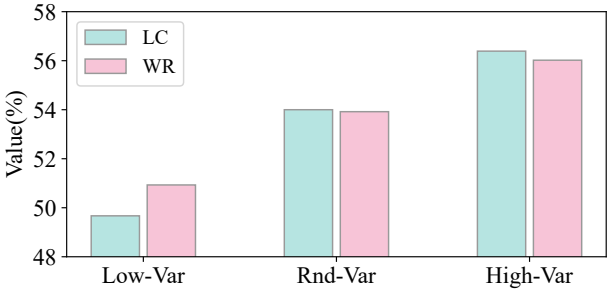


Figure 2: Performance comparison across three different split strategies for Llama-3.2-1B-Instruct on Anthropic HH dataset.

balanced and resilient to noise. RMO begins with a *Dual Denoising Filtering* stage during data preparation, where we identify and filter out ambiguous or noisy sample pairs. This is accomplished by comparing the reward margins predicted by a proxy model (trained on the entire dataset) with those of the base model: samples with decreasing reward margins after training, which indicates low quality or inconsistency, are excluded. In addition, those with persistently small reward margins are probabilistically downsampled. This step improves the overall quality and robustness of the training set. At the batch construction stage, RMO applies *Batch Margin Diversification*, which explicitly maximizes the variance of reward margins within each batch. By ensuring that each batch contains a broad range of margin values, this module provides the model with more diverse and informative preference signals, thereby enhancing generalization and reducing the risk of overfitting to narrow margin distributions. Finally, during training, RMO employs a *Pairwise Margin Amplification* module, which augments the standard training objective with an auxiliary loss term. This term explicitly encourages the model to increase the reward margin between positive and negative samples. By doing so, RMO further strengthens the model's discriminative ability, leading to more effective alignment.

The experimental results show that **RMO consistently outperforms existing preference optimization methods** such as vanilla DPO and SimPO across multiple datasets. Specifically, on models including Llama-3.2 and Pythia (ranging from 1B to 3B parameters), RMO achieves an average improvement of **3.4%** in length-controlled win rate (LC) and **5.2%** in raw win rate (WR) under standard evaluation settings. Ablation studies further demonstrate that each component of RMO contributes significantly to performance gains. Additionally, experiments on the extra dataset and SimPO validate the generalizability of RMO. These results underscore the effectiveness and robustness of reshaping reward margin distributions for enhancing alignment in large language models.

Overall, this work proposes Reward Margin Optimization (RMO), a method that reshapes the reward margin distribution to enhance the preference-based alignment of large language models. RMO consists of three key components: (i) a dual denoising filtering strategy that mitigates the impact of ambiguous and noisy samples during data preparation; (ii) batch margin diversification, which maximizes the balance and diversity of reward margins within each training batch; and (iii) pairwise margin amplification, which explicitly enlarges the reward margin between positive and negative samples during training. Extensive empirical results on multiple baselines demonstrate that RMO consistently improves the alignment performance of LLMs.

Related Work: Alignment of LLMs

Ensuring LLM alignment is essential for making their behavior consistent with human values, intentions, and safety requirements (Ji et al. 2023). A range of alignment methods has been developed to achieve this goal. Proximal Policy Optimization (PPO) leverages reinforcement learning from human feedback (Schulman et al. 2017; Ouyang et al. 2022), while Direct Preference Optimization (DPO) aligns outputs directly through preference comparisons without explicit reward modeling (Rafailov et al. 2023). Response Ranking based on Human Feedback (RRHF) ranks model outputs by conditional probability, achieving PPO-level results with fewer models (Yuan et al. 2023). Implicit Preference Optimization (IPO) approach provides a general framework that bypasses reward modeling and pointwise approximation that bypasses reward modeling. KTO utilizes prospect theory to model human utility, demonstrating superior strong results compared to traditional preference-based strategies, and highlights the importance of inductive bias in human-centric objectives and highlighting inductive bias. SimPO improves DPO by using average log-probabilities as implicit rewards and introducing a margin, achieving better results without a reference model (Meng, Xia, and Chen 2024). Rewards-in-Context (RiC) enables flexible multi-objective alignment by conditioning outputs on reward prompts (Yang et al. 2024). Several works also focus on weak-to-strong alignment (Burns et al. 2023; Guo and Yang 2024; Lyu et al. 2024), or enhance alignment by optimizing prompts (Trivedi et al. 2025; Cheng et al. 2023).

From the data perspective, some methods leverage synthetic data (Xu et al. 2024; Huang et al. 2024) or data selection (Deng et al. 2025; Gao et al. 2025) to improve align-

Algorithm 1: Batch Margin Diversification

Input: Training dataset $\mathcal{D}_{\text{train}} = \{(x_i, y_i^+, y_i^-)\}_{i=1}^N$, number of batches n, batch size $m = N/n$, early stopping threshold ϵ, patience k
Output: Partitioned batches $\{\mathcal{B}_j\}_{j=1}^n$
for $i = 1$ to N **do**
  Compute $\delta_i = \log \pi_\theta(y_i^+ \mid x_i) - \log \pi_\theta(y_i^- \mid x_i)$
end for
Sort $\{\delta_i\}_{i=1}^N$ in ascending order to obtain $\{\delta_{(1)}, \delta_{(2)}, \ldots, \delta_{(N)}\}$
for $j = 1$ to n **do**
  Initialize $\mathcal{B}_j = \{\delta_{(j+kn)} \mid k = 0, 1, \ldots, m-1\}$
end for
Initialize variance improvement buffer $Q \leftarrow []$
while early stopping not triggered **do**
  Randomly select two batches $\mathcal{B}_p, \mathcal{B}_q$
  Randomly select $a \in \mathcal{B}_p$, $b \in \mathcal{B}_q$
  Swap a and b to obtain new batches $\mathcal{B}'_p, \mathcal{B}'_q$
  if Total variance increases **then**
    Accept swap: update $\mathcal{B}_p \leftarrow \mathcal{B}'_p$, $\mathcal{B}_q \leftarrow \mathcal{B}'_q$
    Append improvement to Q
  end if
  if length$(Q) > k$ **and** $\frac{Q[-1]-Q[-k]}{Q[-k]} < \epsilon$ **then**
    break
  end if
end while
Return $\{\mathcal{B}_j\}_{j=1}^n$

ment. However, despite these advances, little attention has been paid to the distributional properties of reward margins during alignment.

Preliminary

Reward Margin. Given a prompt $x \in \mathcal{X}$ and a human preference pair (y^+, y^-) with $y^+ \succ y^-$, let $r : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ denote a scalar reward (or preference) scoring function (e.g., an explicit reward model r_{RM}, or an implicit score derived from model log-probabilities). We define the *reward margin* for the pair as

$$\Delta r(x; y^+, y^-) = r(x, y^+) - r(x, y^-). \tag{1}$$

A large positive Δr indicates a clear preference for y^+ over y^- under r; values near zero reflect ambiguous or low-confidence preferences; and negative values signal disagreement between r and the annotated preference.

Reward Variance. In the context of preference-based alignment, it is often useful to consider the reward variance (Razin et al. 2025) induced by a reward model for a given prompt. Formally, given a policy π_θ, a prompt $x \in \mathcal{X}$, and a reward model $r_{\text{RM}} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, the reward variance induced by r_{RM} for π_θ and x is defined as:

$$\begin{aligned}\text{Var}_{y\sim\pi_\theta(\cdot|x)}\big[r_{\text{RM}}(x,y)\big] = \mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\Big[\big(r_{\text{RM}}(x,y) \\ - \mathbb{E}_{y'\sim\pi_\theta(\cdot|x)}\big[r_{\text{RM}}(x,y')\big]\big)^2\Big].\end{aligned} \tag{2}$$

This quantity measures the diversity of reward signals that the current policy π_θ can obtain for a fixed prompt x under the given reward model.

Preference-based Alignment. Preference-based alignment aims to train models using human feedback in the form of pairwise preferences. This idea has its roots in the classical Bradley-Terry model (Bradley and Terry 1952), which formalizes the probability that one response y^+ is preferred over another y^- under a scoring function r as:

$$P(y^+ \succ y^-) = \frac{\exp(r(x, y^+))}{\exp(r(x, y^+)) + \exp(r(x, y^-))}, \tag{3}$$

where $r(x, y)$ denotes the (possibly learned) utility or reward for response y given prompt x.

DPO. Direct Preference Optimization (DPO) (Rafailov et al. 2023) is a preference-based alignment algorithm that directly trains language models based on human preference data, without relying on explicit reward modeling or reinforcement learning. Given a dataset of preference triples (x, y^+, y^-), where y^+ is the preferred response to prompt x compared to y^-, the DPO objective is formulated as:

$$\begin{aligned}\mathcal{L}_{\text{DPO}} = -\log\sigma\Big(\beta\big(\log\frac{P_\theta(y^+|x)}{P_{\text{ref}}(y^+|x)} \\ - \log\frac{P_\theta(y^-|x)}{P_{\text{ref}}(y^-|x)}\big)\Big),\end{aligned} \tag{4}$$

where P_θ denotes the policy model being optimized, P_{ref} is a reference model, β is a temperature parameter, and $\sigma(\cdot)$ is the sigmoid function. The DPO objective encourages the policy to assign higher probabilities to human-preferred responses, thereby aligning the model more closely with human preferences in a stable and efficient manner.

RMO: Reward Margin Optimization

In this section, we introduce the proposed method—Reward Margin Optimization (RMO). RMO is composed of three key components executed sequentially: **1) Dual Denoising Filtering**, which operates during data preparation by identifying and removing or downsampling ambiguous and noisy sample pairs based on their reward margin statistics both before and after training, thereby improving robustness and the quality of training data; **2) Batch Margin Diversification**, which is applied at the batch construction stage to maximize the variance of reward margins within each batch, encouraging greater diversity and balance in the learning signal; and **3) Pairwise Margin Amplification**, which augments the standard training objective with an auxiliary loss that explicitly increases the reward margin between positive and negative samples, thereby enhancing the model's discriminative capability during training. We elaborate on each component below.

Objective. Given a training dataset $\mathcal{D}_{\text{train}}$ and a testing dataset $\mathcal{D}_{\text{test}}$, our goal is to optimize the parameters θ of the language model such that, after training on the modified dataset and objective induced by RMO, the model π_θ achieves strong alignment performance on $\mathcal{D}_{\text{test}}$. Formally,

RMO seeks to design the data transformation $\mathcal{T}$, batch construction $\mathcal{B}$, and loss function $\mathcal{L}$ such that

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{(x,y^+,y^-)\sim\mathcal{B}(\mathcal{T}(\mathcal{D}_{\text{train}}))} \mathcal{L}(\pi_\theta, x, y^+, y^-) \tag{5}$$

and π_{θ^*} achieves maximal alignment on $\mathcal{D}_{\text{test}}$.

Dual Denoising Filtering

Ambiguous and noisy samples, often characterized by small or negative reward margins, can significantly deteriorate preference alignment. To address this, we propose *Dual Denoising Filtering*, which eliminates unreliable data in two steps:

Step 1: Margin Decrease Filtering. Samples whose reward margin is negative under the base model and further decreases after full-dataset training are likely to be dominated by annotation noise, contradictory labels, or inconsistent human preferences. Retaining such samples can inject harmful supervision and impair the learning of the target alignment objective. Therefore, let δ_i^{base} denote the reward margin for sample i given by the base model, and δ_i^{proxy} the margin given by a proxy model trained on the full set. We identify unreliable samples as:

$$\mathcal{I}_{\text{rm}} = \left\{ i \;\middle|\; \delta_i^{\text{base}} < 0 \;\wedge\; \delta_i^{\text{proxy}} < \delta_i^{\text{base}} \right\} \tag{6}$$

These samples are removed from the training set:

$$\mathcal{D}' = \mathcal{D}_{\text{train}} \setminus \left\{ (x_i, y_i^+, y_i^-) \mid i \in \mathcal{I}_{\text{rm}} \right\} \tag{7}$$

Step 2: Small Margin Downsampling. On the remaining set, many samples may still have reward margins that remain very small even after model training, indicating that the model struggles to learn meaningful preference distinctions for these pairs. To mitigate the influence of such inherently ambiguous pairs, we assign each sample a retention probability according to the absolute proxy margin $x_i = |\delta_i^{\text{proxy}}|$:

$$P_{\text{sample}}(x_i) = 0.5 \left[1 - \tanh\left(\frac{x_i - \tau}{\gamma} \right) \right] \tag{8}$$

where τ and γ are tunable hyperparameters controlling the threshold and scale of the sampling function, respectively. Each sample is retained with probability $P_{\text{sample}}(x_i)$ to construct the final denoised training set:

$$\mathcal{D}_{\text{final}} = \left\{ (x_i, y_i^+, y_i^-) \in \mathcal{D}' \;\middle|\; \text{bernoulli}(P_{\text{sample}}(x_i)) = 1 \right\} \tag{9}$$

Through this two-stage filtering process, unreliable samples are removed or down-weighted, reducing noise and improving the robustness of preference alignment.

Batch Margin Diversification

A fundamental challenge in preference-based alignment is that, due to the natural skewness of reward margin distributions, randomly constructed training batches often lack sufficient diversity. This results in batches dominated by either small or similar reward margins, leading to weaker and less informative optimization signals for the model. To address this, we propose *Batch Margin Diversification*, which aims to explicitly maximize the variance of reward margins within each batch during training.

Algorithm 2: Reward Margin Optimization (RMO)

Input: Original training set $\mathcal{D}_{\text{train}}$, batch size m, number of epochs E, model parameters θ
Output: Optimized model parameters θ^*
// Step 1: Dual Denoising Filtering (Data Preparation)
Compute base and proxy reward margins for each sample in $\mathcal{D}_{\text{train}}$
Remove samples with negative and decreasing margins; downsample samples with small margins)
$\mathcal{D}_{\text{final}} \leftarrow$ resulting filtered training set
// Step 2: Batch Margin Diversification (Batch Construction)
for each epoch $e = 1, \ldots, E$ **do**
    Partition $\mathcal{D}_{\text{final}}$ into n batches of size m using stratified initialization)
    Iteratively refine batch assignments via sample swapping to maximize intra-batch margin variance (Alg. 1)
    // Step 3: Pairwise Margin Amplification (Training)
    for each batch **do**
        For each pair, compute DPO loss $\mathcal{L}_{\text{DPO}}$ and margin δ
        Compute global median margin $\tilde{\delta}$
        Compute auxiliary loss $\mathcal{L}_{\text{reg}} = \lambda_{\text{reg}} \cdot \sigma\left(\frac{\tilde{\delta}-\delta}{\alpha}\right)$
        Update model parameters θ to minimize total loss $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{DPO}} + \mathcal{L}_{\text{reg}}$
    end for
end for
Return θ^*

Given a training dataset $\mathcal{D}_{\text{train}} = \{(x_i, y_i^+, y_i^-)\}_{i=1}^N$ consisting of N preference pairs, we partition the data into n batches, each containing m pairs (so $N = n \times m$). For each batch, we compute the variance of the reward margins, where the reward margin for the i-th pair is defined as $\delta_i = \log \pi_\theta(y_i^+ \mid x_i) - \log \pi_\theta(y_i^- \mid x_i)$. The goal of Batch Margin Diversification is to maximize the sum of the reward margin variances across all batches, thereby ensuring that each batch exhibits as much margin diversity as possible:

$$\max_{\mathcal{B}_1,\ldots,\mathcal{B}_n} \sum_{j=1}^{n} \text{Var}_{i\in\mathcal{B}_j}[\delta_i] \tag{10}$$

where $\mathcal{B}_j$ denotes the set of samples in the j-th batch.

To further improve the efficiency of the optimization, we employ a stratified initialization strategy. Specifically, let $\{\delta_i\}_{i=1}^N$ denote the reward margins of all N training pairs, and let n be the number of batches (so each batch has size $m = N/n$). We first sort the training pairs such that $\delta_{(1)} \le \delta_{(2)} \le \cdots \le \delta_{(N)}$, where $(\cdot)$ denotes the index after sorting.

We then initialize the batch assignments as follows: for $j = 1, \ldots, n$, assign the i-th sample in batch j as

$$\mathcal{B}_j = \left\{ \delta_{(j+kn)} \;\middle|\; k = 0, 1, \ldots, m-1 \right\}. \tag{11}$$

In other words, the j-th batch is composed of the $(j), (j+n), (j+2n), \ldots$-th elements of the sorted margin sequence, ensuring each batch contains a broad spectrum of reward margins.

This initialization ensures that, before any further optimization, the sum of intra-batch variances $\sum_{j=1}^{n} \mathrm{Var}_{i \in \mathcal{B}_j}[\delta_i]$ is already maximized to a certain extent, which significantly accelerates the convergence of subsequent refinement steps.

After this stratified initialization, we further refine the batch assignments to maximize the total variance. Specifically, at each iteration, we randomly select two samples from different batches and propose swapping them. The swap is accepted if and only if it increases the sum of intra-batch variances, i.e.,

$$\sum_{j=1}^{n} \mathrm{Var}_{i \in \mathcal{B}_j^{\text{new}}}[\delta_i] > \sum_{j=1}^{n} \mathrm{Var}_{i \in \mathcal{B}_j^{\text{old}}}[\delta_i]. \tag{12}$$

This local search procedure is repeated iteratively to progressively improve the batch-wise margin diversity. To avoid excessive computation, we introduce an early stopping criterion: the optimization is terminated when the improvement in the total variance over k consecutive iterations falls below a predefined threshold ϵ (e.g., 0.1%).

Pairwise Margin Amplification

Recall that the reward variance induced by a reward model r_{RM} and policy π_θ for a fixed prompt x is defined as in Eq. 2. For the special case where $\pi_\theta(\cdot|x)$ samples only the two preference candidates (y^+, y^-) with equal probability (i.e., a uniform distribution over two options), it is easy to show that:

$$\begin{aligned} \mathrm{Var}_{y \sim \mathrm{Uniform}(y^+, y^-)}\big[r_{\mathrm{RM}}(x,y)\big] &= \frac{1}{4}\big(r_{\mathrm{RM}}(x,y^+) - \\ & r_{\mathrm{RM}}(x,y^-)\big)^2 = \frac{1}{4}\left(\Delta r(x; y^+, y^-)\right)^2, \end{aligned} \tag{13}$$

where $\Delta r(x; y^+, y^-) = r_{\mathrm{RM}}(x, y^+) - r_{\mathrm{RM}}(x, y^-)$ denotes the pairwise reward margin.

This equivalence reveals a direct connection between reward variance and the pairwise reward margin: *maximizing the margin between positive and negative samples increases the reward variance*. Prior work (Razin et al. 2025) has shown that higher reward variance is crucial for training high-quality reward models. In DPO and similar alignment algorithms, the LLM itself acts as the reward model, making it theoretically well-motivated to explicitly enlarge the pairwise margin to enhance the model's alignment performance.

Based on this insight, our *Pairwise Margin Amplification* module introduces an auxiliary loss term to directly encourage the model to increase the reward margin $\Delta r(x; y^+, y^-)$ for each preference pair. Specifically, we augment the standard DPO objective (as shown in Eq. 4) with an auxiliary regularization term. For each training pair, let δ denote the margin for this pair, and let $\tilde{\delta}$ denote the median of all margins globally. The auxiliary loss is defined as:

$$\mathcal{L}_{\mathrm{reg}} = \lambda_{\mathrm{reg}} \cdot \sigma\big(\frac{\tilde{\delta} - \delta}{\alpha}\big), \tag{14}$$

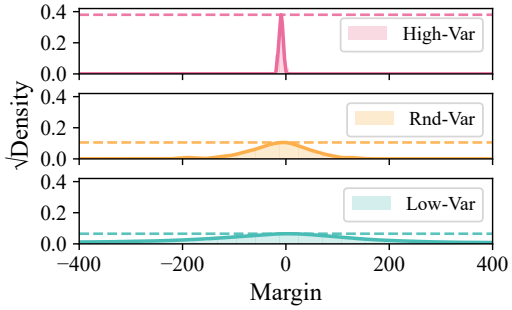


Figure 3: KDE curves of batch-level average margin distributions among different split strategies for Llama-3.2-1B-Instruct.

where α controls the input scale, $\sigma(\cdot)$ is the sigmoid function, and λ_{reg} is a regularization weight. This term penalizes cases where the margin δ for a pair falls below the median margin $\tilde{\delta}$, thus encouraging all pairs to achieve larger margins. The total training objective becomes:

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{DPO}} + \mathcal{L}_{\mathrm{reg}}. \tag{15}$$

Experiment

In this section, we evaluate the performance of the proposed RMO on various models, datasets, and alignment algorithms. We also demonstrate the effectiveness of each component in RMO.

Experiment Setup

Datasets & Benchmarks. We utilize the Anthropic HH dataset (Bai et al. 2022) for our experimental analysis. Considering potential biases that may affect the reliability of our results, we also conduct ablation studies on UltraFeedback dataset (Cui et al. 2024). In each one-step dialogue, a query corresponds to a pair of responses: one deemed more preferred by human annotators and the other less preferred. We randomly sample 10,000 dialogues and split it into 1:4 for SFT and preference optimization individually.

Following β-DPO (Wu et al. 2024), we assess our method using one of the most popular benchmarks for instruction-following models: AlpacaEval 2.0 (Dubois et al. 2025; Zheng et al. 2023). We utilize two metrics: Length-controlled (LC) and raw win rate (WR) to measure how often the GPT-4o model (the recommended replacement for AlpacaEval 2.0, as GPT-4-1106-preview has been officially deprecated) prefers a response generated by a fine-tuned model over the original model. We opt not to rely on the reference provided in this benchmark to facilitate a more direct and fine-grained evaluation of finetuning effects. For more details, please refer to Appendix.

Models. We use two families of models: Llama-3.2 (Dubey et al. 2024) and Pythia (Biderman et al. 2023), each evaluated under two configurations: Instruct and Base. The Llama-3.2 models (Llama-3.2-1B-Instruct and Llama-3.2-3B-Instruct) are pre-trained in an instruction-following setup. For the Pythia models (Pythia-1.4B and Pythia-2.8B),

Method	Llama-3.2-1B-Instruct		Llama-3.2-3B-Instruct		Pythia-1.4B		Pythia-2.8B	
	LC(%)	WR(%)	LC(%)	WR(%)	LC(%)	WR(%)	LC(%)	WR(%)
Low-Var	49.67	$50.93_{\pm 1.76}$	49.70	$48.07_{\pm 1.76}$	56.67	$57.02_{\pm 1.75}$	54.74	$55.09_{\pm 1.75}$
Rnd-Var	54.00	$53.42_{\pm 1.76}$	50.00	$52.79_{\pm 1.76}$	57.09	$57.14_{\pm 1.75}$	55.80	$56.15_{\pm 1.75}$
High-Var	**56.39**	$\mathbf{56.02}_{\pm 1.75}$	**53.81**	$\mathbf{53.42}_{\pm 1.76}$	**59.20**	$\mathbf{60.00}_{\pm 1.73}$	**59.72**	$\mathbf{60.06}_{\pm 1.73}$

Table 1: Performance comparison across different models for three margin variance settings. LC and WR denote length-controlled and raw win rate, respectively.

		Llama-3.2-1B-Instruct		Llama-3.2-3B-Instruct		Pythia-1.4B		Pythia-2.8B	
		LC(%)	WR(%)	LC(%)	WR(%)	LC(%)	WR(%)	LC(%)	WR(%)
Vanilla-DPO		$54.00_{\pm 0.22}$	$53.42_{\pm 1.76}$	$50.00_{\pm 0.14}$	$52.79_{\pm 1.76}$	$57.09_{\pm 0.32}$	$57.14_{\pm 1.75}$	$55.80_{\pm 0.33}$	$56.15_{\pm 1.75}$
Rnd-Var	**+Filter**	$56.05_{\pm 0.19}$	$55.78_{\pm 1.75}$	$53.68_{\pm 0.15}$	$53.42_{\pm 1.76}$	$59.14_{\pm 0.39}$	$59.38_{\pm 1.73}$	$59.16_{\pm 0.39}$	$59.50_{\pm 1.73}$
	+Reg	$54.98_{\pm 0.28}$	$54.78_{\pm 1.75}$	$52.25_{\pm 0.13}$	$52.55_{\pm 1.76}$	$58.75_{\pm 0.37}$	$59.13_{\pm 1.75}$	$57.49_{\pm 0.32}$	$57.39_{\pm 1.75}$
	Rnd-RMO	$57.83_{\pm 0.31}$	$57.40_{\pm 1.74}$	$54.45_{\pm 0.19}$	$54.29_{\pm 1.76}$	$59.96_{\pm 0.41}$	$60.50_{\pm 1.72}$	$59.41_{\pm 0.30}$	$59.75_{\pm 1.73}$
High-Var	**DPO**	$56.39_{\pm 0.19}$	$56.02_{\pm 1.75}$	$53.81_{\pm 0.20}$	$53.42_{\pm 1.76}$	$59.20_{\pm 0.36}$	$60.00_{\pm 1.73}$	$59.72_{\pm 0.32}$	$60.06_{\pm 1.73}$
	+Filter	$56.43_{\pm 0.22}$	$55.65_{\pm 1.75}$	$54.58_{\pm 0.12}$	$54.16_{\pm 1.76}$	$60.78_{\pm 0.33}$	$61.24_{\pm 1.72}$	$60.79_{\pm 0.45}$	$60.93_{\pm 1.72}$
	+Reg	$57.25_{\pm 0.21}$	$56.89_{\pm 1.75}$	$54.02_{\pm 0.34}$	$53.66_{\pm 1.75}$	$62.34_{\pm 0.38}$	$62.48_{\pm 1.71}$	$59.80_{\pm 0.43}$	$60.19_{\pm 1.75}$
	RMO	$\mathbf{59.78}_{\pm 0.22}$	$\mathbf{59.13}_{\pm 1.73}$	$\mathbf{55.72}_{\pm 0.15}$	$\mathbf{55.78}_{\pm 1.75}$	$\mathbf{62.36}_{\pm 0.42}$	$\mathbf{62.61}_{\pm 1.71}$	$\mathbf{61.31}_{\pm 0.44}$	$\mathbf{61.50}_{\pm 1.72}$

Table 2: Main results of RMO on four popular models. We ablate each key design of RMO: (1) High-Var denotes adding the Batch Margin Diversification module compared to Rnd-Var, which randomly partitions the data into batches; (2) +Filter denotes adding the Dual Denoising Filtering module, with 6% filter ratio (τ=14); (3) +Reg denotes adding the Pairwise Margin Amplification module, with λ_{reg}=2 .

we first fine-tune the base versions on the Anthropic HH dataset to obtain the corresponding SFT models, followed by further experiments for preference alignment.

Hyperparameter setting. In During Dual Denoising Filtering, we set different values for τ to ensure consistent filter ratio (varying from 2% to 10%) across models and γ is fixed as 1. In Pairwise Margin Amplification, we set $\tilde{\delta}$ as the median of the reward margin from the proxy model. α is set to 125 for DPO and 5 for SimPO, and λ_{reg} is set in the interval $[0, 3]$, as specified in the following results. Other specific settings for DPO training can be found in the Appendix.

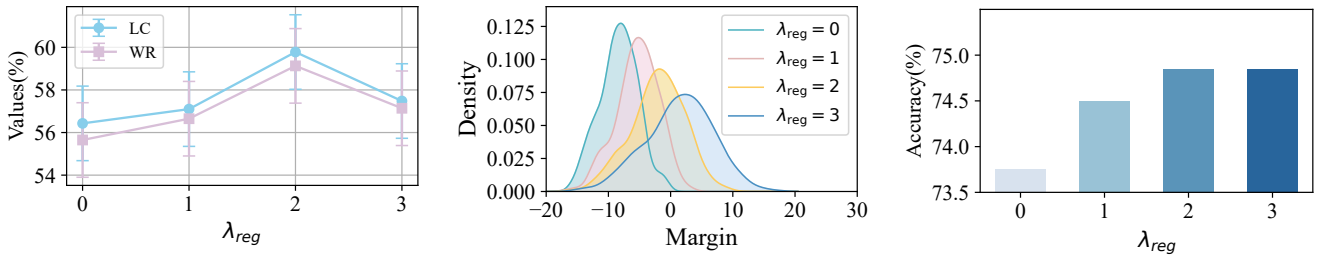
Model	Low-Var	Rnd-Var	High-Var
Pythia-1.4B	523	19108	20492 $_{\uparrow 7.2\%}$
Pythia-2.8B	434	17999	19302 $_{\uparrow 7.2\%}$
Llama-3.2-1B-Instruct	516	20158	21793 $_{\uparrow 8.1\%}$
Llama-3.2-3B-Instruct	623	19164	20509 $_{\uparrow 7.0\%}$

Table 3: Margin variance for different batch partitioning.

Main Results

For each model, we constructed three datasets from the same raw data in the Anthropic HH dataset, using different batch partitioning methods: Low-Var, Rnd-Var, and High-Var. For Rnd-Var, we randomly shuffled the data and constructed the batches. For Low-Var, we use the Batch Margin Diversification algorithm with the objective flipped to minimize, instead of maximize, the margin variance within each batch, serving as an extreme contrast. For High-Var, we apply the original algorithm to maximize the variance of reward margins in each batch. The specific variance values for each method are compared in Table 3.

Diverse margin distribution generally facilitates DPO performance. Figure 3 illustrates the corresponding distributions of batch-level average margin on Llama-3.2-1B-Instruct. RMO reshapes batch composition such that the distribution of batch-level average margins is sharply centered around zero. This indicates a more balanced composition within each batch, reflecting a well-proportioned presence of both positive and negative margins within each batch. In contrast, Low-Var and Rnd-Var show skewed or unstructured distributions. Table 1 demonstrates that performance significantly and consistently improves across all models and evaluation metrics with increasing variance of reward margins, emphasizing the effectiveness and robustness of RMO. Notably, High-Var achieves an average improvement of 4.59% in LC and 4.60% in WR compared to Low-Var, underscoring the critical role of balanced data distribution



(a) Win rate performance with different λ_{reg} (b) Comparisons of margin distributions. (c) Accuracy with different λ_{reg}

Figure 4: Analysis of different regularization intensities on Llama-3.2-1B-Instruct.

in achieving stable optimization. Moreover, runtime analysis in the Appendix confirms the method's high efficiency.

	Anthropic HH		UltraFeedback	
Method	**LC(%)**	**WR(%)**	**LC(%)**	**WR(%)**
Vanilla-SimPO	$60.00_{\pm 0.09}$	$60.62_{\pm 1.72}$	$65.91_{\pm 0.06}$	$64.72_{\pm 1.68}$
+Filter	$61.60_{\pm 0.08}$	$62.32_{\pm 1.70}$	$67.76_{\pm 0.08}$	$67.33_{\pm 1.70}$
+High	$61.44_{\pm 0.07}$	$61.37_{\pm 1.70}$	$67.06_{\pm 0.05}$	$66.21_{\pm 1.67}$
RMO(w/o Reg)	$62.62_{\pm 0.04}$	$64.10_{\pm 1.70}$	$68.32_{\pm 0.05}$	$67.82_{\pm 1.68}$
RMO	$63.76_{\pm 0.05}$	$64.84_{\pm 1.68}$	$69.52_{\pm 0.08}$	$68.82_{\pm 1.63}$

Table 4: RMO Performance on SimPO across different datasets on Llama-3.2-3B-Instruct.

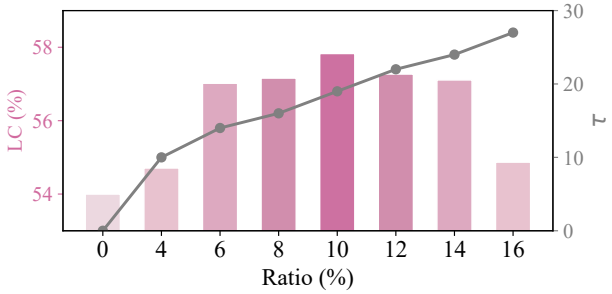


Figure 5: Win rate performance with different filter ratios on Llama-3.2-1B-Instruct.

RMO introduces significant performance gains for DPO. As shown in Table 2, experiments on both Rnd-Var and High-Var settings show that RMO consistently outperforms across all models, with average improvements of 3.4% and 5.2% (LC and WR). High-Var generally yields better results than Rnd-Var, aligning with findings in Table 1. Additionally, ablation studies show that adding Dual Denoising Filtering (+Filter) and Pairwise Margin Amplification (+Reg) independently boosts performance on each dataset, with gains of 3.52% and 3.78% in High-Var, underscoring the effectiveness of these components.

RMO can be transferred to other preference optimization methods and datasets. To expand our approach to more datasets and preference optimization methods, we experiment on one of the most popular methods, SimPO (Meng, Xia, and Chen 2024), and include an extra dataset–UltraFeedback (Cui et al. 2024). As shown in Table 4, RMO and its components demonstrate significant performance improvements on both datasets, which substantiates the robustness of RMO's scalability.

Hyperparameter analysis of regularization and filter strategy. We investigate the effect of varying λ_{reg} in Pairwise Margin Amplification (Eq. 14), assessing margin distribution and performance on AlpacaEval2's LC and WR, alongside accuracy on a given dataset. As shown in Figure 4, increasing λ_{reg} successfully shifts the margin distribution rightward, demonstrating that our regularization strategy effectively improves the distinction between answer pairs, leading to clearer optimization signals. Simultaneously, accuracy improves with λ_{reg}. However, LC and WR do not monotonically increase with λ_{reg}; instead, they peak at λ_{reg}=2. We hypothesize that the interaction between $\mathcal{L}_{\text{DPO}}$ and $\mathcal{L}_{\text{reg}}$ in Equation 15 results in a trade-off: an excessively large $\mathcal{L}_{\text{reg}}$ may overshadow the main loss $\mathcal{L}_{\text{DPO}}$, causing performance degradation.

Figure 5 shows the LC performance for different filter ratios (4% to 16%). In Dual Denoising Filtering, the sample count in **Step 1** (Eq. 6) is fixed and the filter ratio is controlled by the threshold parameter τ in **Step 2** (Eq. 8). As the ratio increases, LC rises steadily and peaks at 10%. Beyond this point, an overly aggressive filter discards informative preference pairs, causing LC to decline.

Conclusion

In this work, we propose RMO, a framework that improves LLM alignment by reshaping reward margin distributions throughout data preparation, batch construction, and training. RMO enhances answer pair distinction and learning signals, leading to consistent performance gains across models and datasets, and scales to various preference optimization algorithms. While RMO introduces extra hyperparameters during filtering and regularization, future work will focus on adaptive adjustment and broader algorithmic applications.

References

Bai, Y.; Jones, A.; Ndousse, K.; Askell, A.; Chen, A.; DasSarma, N.; Drain, D.; Fort, S.; Ganguli, D.; Henighan, T.; Joseph, N.; Kadavath, S.; Kernion, J.; Conerly, T.; El-Showk, S.; Elhage, N.; Hatfield-Dodds, Z.; Hernandez, D.; Hume, T.; Johnston, S.; Kravec, S.; Lovitt, L.; Nanda, N.; Olsson, C.; Amodei, D.; Brown, T.; Clark, J.; McCandlish, S.; Olah, C.; Mann, B.; and Kaplan, J. 2022. Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback. arXiv:2204.05862.

Biderman, S.; Schoelkopf, H.; Anthony, Q.; Bradley, H.; O'Brien, K.; Hallahan, E.; Khan, M. A.; Purohit, S.; Prashanth, U. S.; Raff, E.; Skowron, A.; Sutawika, L.; and van der Wal, O. 2023. Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling. arXiv:2304.01373.

Bradley, R. A.; and Terry, M. E. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4): 324–345.

Burns, C.; Izmailov, P.; Kirchner, J. H.; Baker, B.; Gao, L.; Aschenbrenner, L.; Chen, Y.; Ecoffet, A.; Joglekar, M.; Leike, J.; et al. 2023. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. *arXiv preprint arXiv:2312.09390*.

Cheng, J.; Liu, X.; Zheng, K.; Ke, P.; Wang, H.; Dong, Y.; Tang, J.; and Huang, M. 2023. Black-box prompt optimization: Aligning large language models without model training. *arXiv preprint arXiv:2311.04155*.

Cui, G.; Yuan, L.; Ding, N.; Yao, G.; He, B.; Zhu, W.; Ni, Y.; Xie, G.; Xie, R.; Lin, Y.; Liu, Z.; and Sun, M. 2024. UltraFeedback: Boosting Language Models with Scaled AI Feedback. arXiv:2310.01377.

Deng, X.; Zhong, H.; Ai, R.; Feng, F.; Wang, Z.; and He, X. 2025. Less is more: Improving llm alignment via preference data selection. *arXiv preprint arXiv:2502.14560*.

Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; et al. 2024. The llama 3 herd of models. *arXiv e-prints*, arXiv–2407.

Dubois, Y.; Galambosi, B.; Liang, P.; and Hashimoto, T. B. 2025. Length-Controlled AlpacaEval: A Simple Way to Debias Automatic Evaluators. arXiv:2404.04475.

Ethayarajh, K.; Xu, W.; Muennighoff, N.; Jurafsky, D.; and Kiela, D. 2024. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*.

Gao, C.; Li, H.; Liu, L.; Xie, Z.; Zhao, P.; and Xu, Z. 2025. Principled data selection for alignment: The hidden risks of difficult examples. *arXiv preprint arXiv:2502.09650*.

Guo, Y.; and Yang, Y. 2024. Improving weak-to-strong generalization with reliability-aware alignment. *arXiv preprint arXiv:2406.19032*.

Hadi, M. U.; Qureshi, R.; Shah, A.; Irfan, M.; Zafar, A.; Shaikh, M. B.; Akhtar, N.; Wu, J.; Mirjalili, S.; et al. 2023. A survey on large language models: Applications, challenges, limitations, and practical usage. *Authorea Preprints*.

Huang, Y.; Wu, S.; Gao, C.; Chen, D.; Zhang, Q.; Wan, Y.; Zhou, T.; Xiao, C.; Gao, J.; Sun, L.; et al. 2024. Datagen: Unified synthetic dataset generation via large language models. In *The Thirteenth International Conference on Learning Representations*.

Ji, J.; Qiu, T.; Chen, B.; Zhang, B.; Lou, H.; Wang, K.; Duan, Y.; He, Z.; Zhou, J.; Zhang, Z.; et al. 2023. Ai alignment: A comprehensive survey. *arXiv preprint arXiv:2310.19852*.

Lyu, Y.; Yan, L.; Wang, Z.; Yin, D.; Ren, P.; de Rijke, M.; and Ren, Z. 2024. MACPO: weak-to-strong alignment via multi-agent contrastive preference optimization. *arXiv preprint arXiv:2410.07672*.

Meng, Y.; Xia, M.; and Chen, D. 2024. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37: 124198–124235.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744.

Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C. D.; Ermon, S.; and Finn, C. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36: 53728–53741.

Razin, N.; Wang, Z.; Strauss, H.; Wei, S.; Lee, J. D.; and Arora, S. 2025. What makes a reward model a good teacher? an optimization perspective. *arXiv preprint arXiv:2503.15477*.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Trivedi, P.; Chakraborty, S.; Reddy, A.; Aggarwal, V.; Bedi, A. S.; and Atia, G. K. 2025. Align-pro: A principled approach to prompt optimization for llm alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 27653–27661.

Wu, J.; Xie, Y.; Yang, Z.; Wu, J.; Gao, J.; Ding, B.; Wang, X.; and He, X. 2024. β-DPO: Direct Preference Optimization with Dynamic β. *Advances in Neural Information Processing Systems*, 37: 129944–129966.

Xu, Z.; Jiang, F.; Niu, L.; Deng, Y.; Poovendran, R.; Choi, Y.; and Lin, B. Y. 2024. Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing. *arXiv preprint arXiv:2406.08464*.

Yang, R.; Pan, X.; Luo, F.; Qiu, S.; Zhong, H.; Yu, D.; and Chen, J. 2024. Rewards-in-Context: Multi-objective Alignment of Foundation Models with Dynamic Preference Adjustment. In *Forty-first International Conference on Machine Learning*.

Yuan, H.; Yuan, Z.; Tan, C.; Wang, W.; Huang, S.; and Huang, F. 2023. Rrhf: Rank responses to align language models with human feedback. *Advances in Neural Information Processing Systems*, 36: 10935–10950.

Zheng, L.; Chiang, W.-L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E.; et al. 2023.

Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36: 46595–46623.

Zhou, H.; Hu, C.; Yuan, Y.; Cui, Y.; Jin, Y.; Chen, C.; Wu, H.; Yuan, D.; Jiang, L.; Wu, D.; Liu, X.; Zhang, J.; Wang, X.; and Liu, J. 2025. Large Language Model (LLM) for Telecommunications: A Comprehensive Survey on Principles, Key Techniques, and Opportunities. *IEEE Communications Surveys & Tutorials*, 27(3): 1955–2005.