RSPO: Reward-Driven Selective Penalization for Preference Alignment Optimization

Anonymous ACL submission

Abstract

Preference optimization is a crucial research direction for aligning language models with human preferences. Direct Preference Optimization (DPO) has emerged as a novel approach, replacing the paradigm of Reinforcement Learning from Human Feedback (RLHF) with the direct optimization of preference reward functions. However, DPO treats all preference response pairs as equally important, regardless of their quality or complexity. This may inadvertently lead to suboptimal generalization, especially when the training dataset is dominated by noisy or ambiguous preference response pairs. In this paper, we propose a novel method called **R**eward-Driven **S**elective **P**enalization for Preference Alignment **O**ptimization (**RSPO**). RSPO for the first time proposes to dynamically categorize preference data based on implicit reward signals and apply selective weighting to different categories. Moreover, RSPO introduces a Penalty Weighting Strategy that dynamically evaluates data quality and adjusts optimization weights in real time, effectively tackling challenges posed by noisy and complex preference signals, thereby improving alignment performance.

Our experiments demonstrate that RSPO achieves remarkable performance on both the Mistral-base and Llama3-Instruct models, outperforming DPO by an average of 4.55% on AlpacaEval 2, and surpassing recent methods such as SimPO and WPO, achieving state-of-the-art (SOTA) performance. Our code is available at `https://anonymous.4open.science/r/RSPO`.

1 Introduction

Preference alignment optimization has recently become a pivotal research focus in the development of large language models (LLMs) (Ouyang et al., 2022; Ziegler et al., 2019; Bai et al., 2022), striving to better align model-generated content with human preferences which plays a crucial role in enhancing user experience and improving performance on downstream tasks (Raffel et al., 2020; Brown et al., 2020; Hoffmann et al., 2022). Traditional training methods, such as unsupervised or semi-supervised learning, rely on fixed objective functions that fail to capture the dynamic and diverse nature of human preferences. As a result, there is a growing need to explore how optimized preference signals can improve model alignment and better accommodate user needs (Askell et al., 2021).

Direct Preference Optimization (DPO) (Rafailov et al., 2023) integrates reward function learning and policy optimization to directly maximize the generation probability of preferred responses, offering a unified framework for preference alignment. However, despite its advantages, DPO faces several inherent limitations that hinder its effectiveness in applications. *One of the key challenges* lies in its uniform treatment of preference data, where all samples are optimized using the same strategy regardless of their alignment quality or complexity. Dynamic reweighting strategies have been shown to be highly effective in enhancing alignment performance. For instance, Xu et al. (2024a) employ curriculum learning (Bengio et al., 2009) to optimize the model's learning trajectory by incrementally introducing data based on sample difficulty and quality. Weighted Preference Optimization (WPO) (Zhou et al., 2024) reweights preference pairs according to their probability under the current policy, thereby improving alignment in LLMs. In contrast, DPO's static optimization approach struggles to fully leverage the distinctions among data samples, failing to dynamically adjust to varying data quality and complexity (Wu et al., 2024b).

Additionally, DPO struggles with noisy or inconsistent annotations common in subjective human evaluations (Wu et al., 2024a; Lee et al., 2023; Gupta et al., 2022). It is also prone to policy distribution shift, where penalizing dispreferred responses may inadvertently impair the generation of preferred ones. Furthermore, DPO can overfit on limited or unrepresentative data, leading to poor generalization on unseen inputs (Xu et al., 2024c).

To overcome these limitations, we propose **RSPO** (**R**eward-Driven **S**elective **P**enalization for Preference Alignment **O**ptimization), a method that dynamically classifies preference data based on implicit reward signals and applies selective weighting to improve alignment. Based on in-depth exploration and analysis, RSPO divides data into well-aligned ($R1$) and less-aligned ($R2$–$R4$) categories, assigning higher weights to $R1$ to reinforce reliable learning while penalizing $R2$–$R4$ to reduce overfitting. Additionally, we propose a novel dynamic Penalty Weighting Strategy, which adjusts optimization weights in real time based on data quality, thereby enhancing stability and robustness in complex alignment scenarios.

Our contributions are summarized as follows:

- We, for the first time, classify preference data into four reward-alignment categories ($R1$–$R4$) based on implicit reward distributions. By leveraging this categorization, RSPO optimizes preference data selectively, assigning higher weights to well-aligned ($R1$) data while in a fine-grained manner adjusting the weights of less-aligned ($R2$–$R4$) samples to enhance robustness and alignment performance.

- We introduce a novel Penalty Weighting Strategy, which dynamically evaluates sample quality and assigns penalties to noisy or hard-to-align data during training. This mechanism improves optimization stability, mitigates overfitting, and ensures effective learning of high-quality preference signals.

- Extensive experimental results demonstrate the effectiveness of RSPO, which is implemented on two widely used large language model foundations: Mistral-base and Llama3-Instruct. On Mistral-base model, RSPO outperforms DPO by 4.8% on the AlpacaEval 2 benchmark. On Llama3-Instruct model, RSPO outperforms DPO by 4.3%. RSPO also surpasses recent methods such as SimPO (Lu et al., 2024) and WPO (Zhou et al., 2024), achieving new SOTA results, and exhibits strong generalization capabilities across multiple downstream tasks.

2 Related Work

Direct Preference Optimization (DPO) and Its Analysis. DPO (Rafailov et al., 2023) has become a prominent method for aligning language models with human preferences by directly optimizing the preference reward function. However, recent studies highlighted several limitations, including imbalanced gradient updates, which hinder its ability to effectively capture complex preference signals (Feng et al., 2024). Additionally, DPO is prone to generating out-of-distribution (OOD) responses and exhibits high sensitivity to distribution shifts between the training dataset and model-generated outputs (Gan et al., 2024), posing challenges for real-world deployment.

Advancements in DPO. To address the limitations of DPO, several methods have been proposed. Smaug (Pal et al., 2024) employs DPO-Positive to fix some failure modes of DPO. WPO (Zhou et al., 2024) adapts off-policy data to resemble on-policy data by reweighting preference pairs according to their probability under the current policy. R-DPO (Park et al., 2024) incorporates an extra regularization term into the DPO loss function, thereby mitigating the model's tendency to exploit text length during the optimization process.

Other Preference Optimization Methods. Various other preference optimization methods have been explored. SimPO (Meng et al., 2024) performs simple preference optimization using a reference-free reward function. GPO (Generalized Preference Optimization) (Tang et al., 2024) offers a unified approach to offline alignment. ORPO (Hong et al., 2024) performs monolithic preference optimization without a reference model. RRHF (Yuan et al., 2023) aligns language models with human preferences by scoring sampled responses from various sources and learning to rank them using ranking loss. CPO (Xu et al., 2024b) uses contrastive learning to compare pairs of outputs and optimize preferences.

3 Method

In this section, we provide the preliminaries of DPO in Section 3.1. We then introduce our proposed RSPO, which primarily consists of two parts, as shown in Figure 1: Classification of DPO Implicit Rewards (Section 3.2) and Reward-Driven Selective Penalization Weighting (Section 3.3).

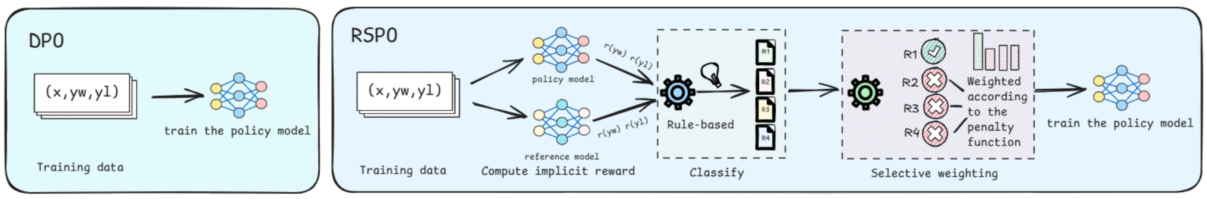


Figure 1: Overview of DPO and our RSPO method. RSPO dynamically classifies preference data based on implicit reward signals into well-aligned $R1$ data and less-aligned $R2$-$R4$ data. It then applies weighting to selected options according to our Penalty Weighting Strategy to evaluate data quality and adjust optimization weights in real time. By assigning higher weights to $R1$ data with clear reward signals, RSPO reinforces the learning of high-alignment data while selectively penalizing $R2$-$R4$ data to mitigate overfitting caused by hard-to-align samples.

3.1 Preliminaries

DPO (Rafailov et al., 2023) unifies reward learning and policy optimization by implicitly reparameterizing the reward function via the closed-form solution of the optimal policy. Given a human preference dataset $\mathcal{D}$, where each pair consists of a preferred response y_w and a dispreferred response y_l, DPO optimizes the policy model π_θ to increase the probability of preferred responses while decreasing the probability of generating dispreferred responses by increasing the log probability of preferred responses over dispreferred responses:

$$\beta \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)},$$

where π_{ref} denotes a reference model.

DPO employs Bradley-Terry (Bradley and Terry, 1952) model to measure the alignment between policy model and human preference. By rearranging the optimal solution of the general reward function, we can derive the following reward function:

$$r^*(x, y) = \beta \log \frac{\pi^*(y|x)}{\pi_{\text{ref}}(y|x)} + \beta \log Z(x),$$

where $Z(x)$ is a normalization factor. By substituting this reward function into the Bradley-Terry model, we can obtain the DPO loss function:

$$L_{\text{DPO}} = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \left[\log \sigma(r_w - r_l) \right]$$

3.2 Classification of DPO Implicit Rewards

In DPO, all samples are optimized using the same strategy, regardless of their alignment quality or complexity. To fully leverage the distinctions among data samples and adjust optimization weights in real time, we innovatively propose dynamically categorizing preference data based on implicit reward signals into well-aligned $R1$ data and less-aligned $R2$-$R4$ data, which is then applied to subsequent preference optimization.

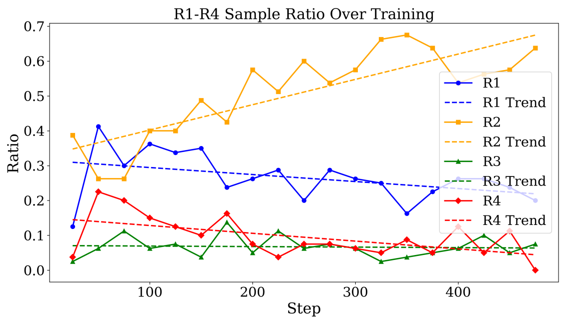
Four Implicit Rewards in DPO. The gradient of L_{DPO} with respect to the parameters θ can be written as:

$$\nabla_\theta L_{\text{DPO}} = -\beta \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \Big[\sigma(r_\theta(x, y_l) - r_\theta(x, y_w)) \cdot (\nabla_\theta \log \pi_\theta(y_w|x) - \nabla_\theta \log \pi_\theta(y_l|x)) \Big],$$

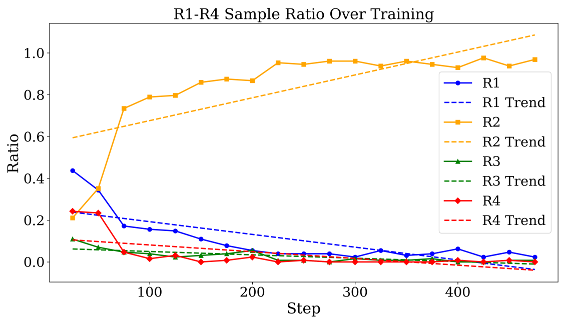
where $r_\theta(x, y) = \beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}$ is the implicit reward defined by the policy model π_θ and reference model π_{ref}. This gradient emphasizes the objective of DPO, which is to dynamically adjust the generation probabilities $p_\theta(x, y_w)$ and $p_\theta(x, y_l)$ based on implicit rewards derived from the preference response pairs. Since the implicit rewards reflect the alignment quality between the policy model and human preferences and serve as a crucial part of the DPO optimization process, we argue that a more detailed study of the preference response pairs and their implicit rewards is necessary.

Based on the implicit rewards $r_\theta(x, y_w)$ and $r_\theta(x, y_l)$, we categorize preference response pairs into four types:

- $R1$: The preferred response exhibits a positive reward ($r_\theta(x, y_w) \geq 0$), while the dispreferred response displays a negative reward ($r_\theta(x, y_l) \leq 0$), representing a clear alignment between policy model and human preferences.
- $R2$: Both preferred and dispreferred responses receive negative reward signals ($r_\theta(x, y_w) < 0$, $r_\theta(x, y_l) \leq 0$), indicating a failure to reinforce preference alignment effectively.
- $R3$: The preferred response receives a negative reward signal, while the dispreferred response gets a positive signal ($r_\theta(x, y_w) < 0$, $r_\theta(x, y_l) > 0$), indicating a misalignment



(a) Mistral-SFT.



(b) Llama-3-Instruct.

Figure 2: Distribution of four types of preference response pairs during DPO training on Mistral-SFT and Llama-3-Instruct. Results for one epoch on the Ultrafeedback dataset (Cui et al., 2023).

where dispreferred responses are erroneously prioritized.

- $R4$: Both preferred and dispreferred responses receive positive reward signals ($r_\theta(x, y_w) \geq 0$, $r_\theta(x, y_l) > 0$), reflecting an ambiguity in the alignment signals.

These four types of preference response pairs exhibit distinct distributions as training progresses. As illustrated in Figure 2, the distribution of preference response pairs indicates that the policy model struggles to consistently capture preference signals for most training data (where $R2$ is the data with the highest proportion in the binarized Ultrafeedback dataset). However, DPO's optimization framework treats all preference response pairs as equally important, regardless of their quality or complexity. This may inadvertently lead to suboptimal generalization, especially when the training dataset is dominated by noisy or ambiguous preference response pairs. Moreover, their distribution proportions vary throughout the entire training process (see **Appendix D**).

Impact of $R1$–$R4$ on the Training Process. In our data classification, we analyze and reveal that the $R1$ type provides a strong alignment signal, indicating that the pairs are well-aligned with the model's learned preference. In contrast, the types $R2$ - $R4$ may exhibit ambiguous, complex, or conflicting situations with respect to the human preferences that the model has learned. Building on the concept of curriculum learning, we believe that selectively reducing the loss for $R2$-$R4$ preferences can help mitigate overfitting on challenging data and enhance the model's ability to generalize human preferences. This finding is experimentally verified in **Section 5.2**.

3.3 Reward-Driven Selective Penalization Weighting

In order to mitigate model's overfitting to preference responce pairs that are difficult to learn preferences and consolidate the preferences that the policy model has learned, we propose a novel reward-driven selective penalization weighting approach, which dynamically reweights these preference response pairs to maximize model performance.

On the basis of observation in Section 3.2, the reweighting strategy should ensure that preference response pairs of $R1$ type retain their full contribution to the optimization objective, reinforcing the model's learned human preferences. In contrast, for preference response pairs of $R2$-$R4$ type, their weights are dynamically adjusted to reflect their relative uncertainty or conflict with the model's learned human preferences. This approach prevents overfitting to challenging or ambiguous preference response pairs while maintaining alignment with the primary optimization objective. Formally, the selective weight $w(y_w, y_l, x)$ is defined as follows:

$$w(y_w, y_l, x) = \begin{cases} f(y_w, y_l, x), & \text{for } R2 - R4 \\ 1, & \text{for } R1 \end{cases} \tag{1}$$

Here, $w(y_w, y_l, x) = 1$ for the preference response pairs of $R1$. In contrast, for $R2$-$R4$, the weight $f(y_w, y_l, x)$ serves as a *penalty function*. The design principle of the penalty function is to reduce the influence of samples on gradient updates based on their deviation from the human preferences learned by the policy model. The penalty weight we define is as follows:

$$f(y_w, y_l, x) = \lambda + \sigma\left(\beta_w \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)}\right) \cdot \sigma\left(\beta_l \log \frac{\pi_{\text{ref}}(y_l|x)}{\pi_\theta(y_l|x)}\right) \tag{2}$$

The design of $f(y_w, y_l, x)$ incorporates two key components:

1. **Constant Scaling Coefficient**: The constant λ $(0 < \lambda < 1)$ provides a baseline penalty to all samples in $R2$-$R4$, ensuring that their influence on the loss is consistently reduced. This mitigates the risk of overfitting to samples with high uncertainty or conflicting signals.

2. **Preference Deviation Penalty Coefficient**: The dynamic penalty coefficient, governed by the sigmoid functions $\sigma(\cdot)$, adjusts the sample weight based on the relative log-probability differences between the policy model π_θ and the reference model π_{ref}. Specifically, the $\sigma\left(\beta_w \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)}\right)$ penalizes deviations in the preferred responses y_w, while $\sigma\left(\beta_l \log \frac{\pi_{\text{ref}}(y_l|x)}{\pi_\theta(y_l|x)}\right)$ penalizes deviations in the rejected responses y_l. This ensures that the penalty is proportionate to the degree of misalignment, suppressing over-learning of ambiguous data and maintaining balance in the preference distribution.

By combining these two components, the penalty weight $f(y_w, y_l, x)$ enables the model to fine-tune its learning process, focusing on clear and reliable data while cautiously handling ambiguous data. The complete RSPO loss function is defined as:

$$L_{\text{RSPO}} = -\mathbb{E}_{(y_w, y_l, x)}[w(y_w, y_l, x) \cdot \log \sigma(r_\theta(x, y_w) - r_\theta(x, y_l))] \quad (3)$$

The detailed steps of the RSPO algorithm are presented in **Appendix A**.

4 Experiment Setup

4.1 Experimental Settings

We utilize Mistral-base[1] and Llama-3-Instruct[2] as our foundational models. For Mistral-base, we utilize the official supervised fine-tuned (SFT) checkpoint from zephyr as our SFT model, training for one epoch on the binarized Ultrafeedback dataset (Cui et al., 2023). For Llama-3-Instruct, we utilize off-the-shelf instruction-tuned model as our SFT model, training for one epoch on the Llama3-Ultrafeedback dataset (Meng et al., 2024). We conduct experiments under both full parameter fine-tuning and LoRA (Hu et al., 2022) fine-tuning settings using identical parameters. More details of experimental settings are provided in **Appendix C**.

[1]https://huggingface.co/HuggingFaceH4/mistral-7b-sft-beta

[2]https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct

4.2 Evaluation

We evaluate the instruction-following capabilities of our models using the widely adopted AlpacaEval 2 benchmark (Li et al., 2023), reporting both the *raw Win Rate* (WR) and the *Length-Controlled Win Rate* (LC) (Dubois et al., 2024). To ensure fair comparison, we follow the decoding strategy of Meng et al. (2024).

In addition, we assess performance on several downstream tasks—MMLU (Hendrycks et al., 2021), ARC (Clark et al., 2018), TruthfulQA (Lin et al., 2022), IFEval (Zhou et al., 2023), and GSM8K (Cobbe et al., 2021)—as detailed in **Appendix B**. We also report standard deviations in **Appendix G** to assess result stability.

4.3 Baselines

We compare our method with recent SOTA preference optimization methods, which are described in detail in Section 2, including RRHF (Yuan et al., 2023), DPO (Rafailov et al., 2023), ORPO (Hong et al., 2024), IPO (Azar et al., 2024), CPO (Xu et al., 2024b), KTO (Ethayarajh et al., 2024), R-DPO (Park et al., 2024), WPO (Zhou et al., 2024), and SimPO (Meng et al., 2024).

5 Experimental Results

5.1 Main Results

Under *full parameter* settings, Table 1 present the results for Mistral-base and Llama-3-Instruct. Our method *consistently outperform all baselines* on AlpacaEval 2: (***i***) In Mistral-base configuration, our RSPO method achieves an LC win-rate of 25.4% and a raw win-rate of 23.7%, improving upon DPO by **4.8**% in LC and **5.5**% in raw win-rate. (***ii***) In Llama-3-Instruct configuration, our method achieves an LC win-rate of 45.0% and a raw win-rate of 42.5%, surpassing DPO by **4.3**% in LC win-rate and **4.4**% in raw win-rate. Moreover, RSPO also outperforms recent SOTA methods SimPO and WPO, achieving new SOTA results.

On multiple downstream tasks, the results in Table 2 show that our method RSPO outperforms DPO across multiple downstream tasks, demonstrating consistent performance improvements. Specifically, compared to DPO, our method

Method	Mistral-base (7B)			Llama-3-Instruct (8B)		
	LC	WR	Avg. Len	LC	WR	Avg. Len
SFT	8.4	6.2	914	26.0	25.3	1920
RRHF (Yuan et al., 2023)	11.6	10.2	1630	31.3	28.4	1805
IPO (Azar et al., 2024)	11.8	9.4	1380	35.6	35.6	1983
KTO (Ethayarajh et al., 2024)	13.1	9.1	1144	33.1	31.8	1909
CPO (Xu et al., 2024b)	9.8	8.9	1827	28.9	32.2	2166
ORPO (Hong et al., 2024)	14.7	12.2	1475	28.5	27.4	1888
DPO (Rafailov et al., 2023)	20.6	18.2	1521	40.7	38.1	1933
R-DPO (Park et al., 2024)	17.4	12.8	1335	41.1	37.8	1854
SimPO (Meng et al., 2024)	21.5	20.8	1868	44.7	40.5	1825
WPO (Zhou et al., 2024)	24.4	**23.7**	-	40.0	41.9	2084
RSPO	**25.4**	**23.7**	1873	**45.0**	**42.5**	1870

Table 1: Comparison of methods on Mistral-base (7B) and Llama-3-Instruct (8B) on AlpacaEval 2 judged by GPT-4-turbo. Each column's maximum value is bolded. "Avg. Len" denotes the average number of output tokens.

Method	GSM8K	ARC	TQA	MMLU	IFEval	Avg.
SFT	**42.61**	55.97	28.15	57.17	36.59	44.10
DPO	33.13	59.64	46.14	57.46	50.48	49.37
R-DPO	30.10	56.06	40.64	58.48	53.24	47.70
SimPO	33.59	**60.15**	43.45	58.25	52.98	49.68
WPO	30.63	57.00	40.51	58.54	**55.64**	48.46
RSPO	37.45	57.94	**47.25**	**58.58**	55.04	**51.25**

Table 2: Performance comparison of different methods on Mistral-Base (7B) across multiple benchmarks (TQA indicates TruthfulQA). We report the strictly match accuracy, and compare with the methods that achieve similar performance to ours on AlpacaEval 2. "Avg" denotes the average performance of all tasks.

	Mistral-base		Llama-3-Instruct	
	DPO	RSPO	DPO	RSPO
Armo WR	20.7	**27.2**	67	**74.2**
Deepseek LC	20.5	**25.9**	47.8	**49.4**
Deepseek WR	16.0	**24.2**	45.6	**45.9**

Table 3: Results of AlpacaEval 2 judged by Armo Llama3 and DeepSeek v3 under the LoRA settings.

achieves higher scores on tasks such as GSM8K, TruthfulQA, and IFEval. Additionally, our method *achieves the highest average score* 51.25% among all methods, further demonstrating its effectiveness.

Under *LoRA* fine-tuning settings, RSPO also consistently outperforms DPO across both Mistral-base and Llama-3-Instruct. The results in Table 3 highlight RSPO's superior performance and demonstrate that our method achieves significant improvements with fewer parameter adjustments.

5.2 Analysis on Implicit Reward Classification

To explore the impact of classifying preference response pairs based on implicit rewards during training on model performance, we conduct the following two sets of experiments:

1. Without distinguishing data based on implicit rewards, all data are given the penalty weight from Equation (2) (named RSPO$_{NoClass}$);

2. Distinguishing data based on implicit rewards, and $R2$-$R4$ data are given the penalty weight from Equation (2) (i.e., our RSPO).

We utilize Mistral-base as the base model, conducting a single epoch of training on the binarized Ultrafeedback dataset (Cui et al., 2023) and evaluating on AlpacaEval 2. The same configuration is employed for the subsequent ablation studies.

The experimental results are shown in Table 4. *First*, the results show that RSPO outperforms RSPO$_{NoClass}$ across all metrics. This suggests that categorizing data into different reward types and applying penalty weights to $R2$-$R4$ data help the model capture human preferences more accurately, enhancing overall training effectiveness. *Second*, both RSPO$_{NoClass}$ and RSPO outperform DPO. This further demonstrates the effectiveness of our

	ARMO	DeepSeek v3	
Method	WR	LC	WR
DPO	20.7	20.5	16.0
RSPO$_{NoClass}$	24.8	23.4	22.6
RSPO	**27.2**	**25.9**	**24.2**

Table 4: Comparison of the impact of preference response pair classification on model performance.

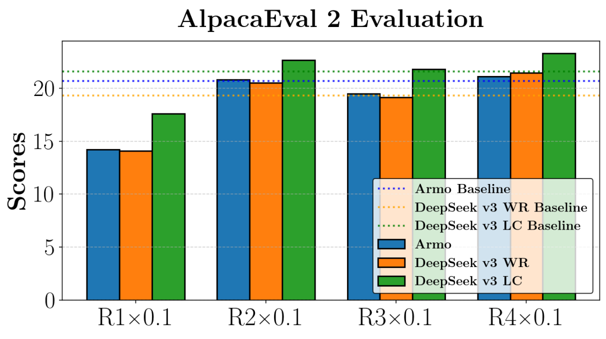


Figure 3: Exploratory experiments: Impact of $R1$–$R4$ on the training process when scaling the loss of each type of preference response pairs by a factor of 0.1.

penalty weighting, which assigns weights based on the degree of preference divergence between the data and the model, as reflected by the model's implicit rewards.

To further investigate the impact of different preference response pairs on training, we conduct exploratory experiments by applying a weight decay factor of 0.1 to each type and using the DPO loss. As shown in Figure 3, applying weight decay to preference response pairs of $R1$ type result in a significant degradation in model alignment performance compared to the DPO baseline. In contrast, applying weight decay to the preference response pairs of $R2$ - $R4$ types lead to alignment performance comparable to or even better than the DPO baseline. These results indicate that different types of preference response pairs exert distinct influences on model training. Notably, reducing the weight of preference pairs of $R1$ type appears to have a significant negative impact on performance.

5.3 Analysis on Penalty Weights

To verify the impact of penalty weights on performance, we conduct the following experiments.

Experimental Analysis of Penalty Weight Range. To validate our hypothesis that reducing the loss on $R2$–$R4$ samples mitigates overfitting and improves

	ARMO	DeepSeek v3	
	WR	LC	WR
λ= 0.075	21.6	21.2	18.6
λ= 1	20.7	20.5	16.0
λ= 5	19.3	15.0	10.8
λ= 10	12.2	12.4	9.4

Table 5: Performance comparison for different penalty weight values of λ.

generalization to human preferences, we adjust the penalty weights as follows:

$$w(y_w, y_l, x) = \begin{cases} \lambda, & \text{for } R2\text{–}R4 \\ 1, & \text{for } R1 \end{cases}$$

where $0 < \lambda < 1$. Lowering λ reduces the impact of $R2$–$R4$ samples during training, while $\lambda > 1$ increases their contribution. As shown in Table 5, appropriately reducing λ improves model performance, whereas increasing it significantly degrades results. This supports our approach of down-weighting $R2$–$R4$ data to consolidate learned preferences and enhance generalization.

Empirically, we set $\lambda = 0.075$, based on the observation that $R1$ samples comprise only about 10% of training data. When $\lambda > 0.2$, the relative weight of $R1$ remains insufficient. By reducing the $R2$–$R4$ weights to approximately 10%, we effectively emphasize $R1$'s contribution during training.

Experimental Analysis of Penalty Weight Function. To assess the impact of penalty weighting functions on model performance, we design several penalty weighting functions as shown in Table 10 of **Appendix F**, where we also analyze our design ideas and conduct the hyperparameter exploration for the proposed penalty functions.

The experimental results using three different penalty weight functions, f_1, f_2 and f_3, are presented in Table 6. These results show that different penalty weight functions have varying impacts on model training effectiveness, with our proposed heuristic penalty function yielding superior performance.

5.4 Further Analysis on Why R1 is Better for LLM Alignment

To further investigate why $R1$ samples are more beneficial for LLM alignment, we conduct experimental analyses from two perspectives:

	ARMO	DeepSeek v3	
Function	WR	LC	WR
f_1	25.3	24.1	21.2
f_2	22.6	23.6	20.7
f_3	27.2	25.9	24.2

Table 6: Performance comparison of different penalty weight functions. f_3 is the one proposed in Eq. (2).

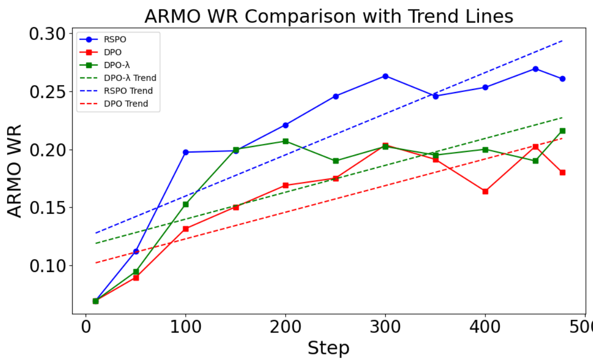


Figure 4: Performance comparison of RSPO, DPO-λ, and DPO (ARMO WR) across training steps (0-450). Dashed lines indicate trend lines.

The evolution of model performance over training steps. In this study, we compare DPO-λ (with $\lambda = 0.075$), RSPO, and standard DPO. DPO-λ applies a constant penalty to $R2$–$R4$ samples, while RSPO employs a dynamic penalty mechanism. Standard DPO applies no explicit penalty. Figure 4 presents the performance of DPO-λ, RSPO, and the standard DPO across different training steps. The results highlight that assigning higher weights to $R1$ samples leads to improved model training. As shown in the figure, both DPO-λ and RSPO achieve better performance more quickly compared to DPO during training. Moreover, RSPO reaches a higher performance ceiling than DPO-λ, demonstrating the superior effectiveness of our proposed RSPO method.

The behavior of model gradients throughout the training process. The gradient-based analysis in Figure 5 reveals that RSPO consistently generates smoother and more stable gradients compared to DPO. This indicates that RSPO not only mitigates abrupt fluctuations in gradient magnitudes but also promotes a more stable and controlled optimization trajectory during training. Such smoothness in gradients is closely associated with enhanced convergence behavior and reduced risk of exploding

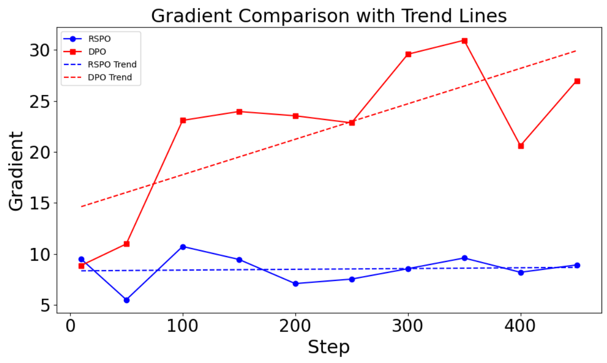


Figure 5: Comparison of RSPO and DPO gradients across steps (0-450).

Metric	**RSPO**	**DPO**	**Diff.**	**Perf. Diff. (LC)**
Avg. GPU Mem. (GB)	436.0	432.9	+3.1	+4.8% (Mistral), +4.3% (Llama)
Avg. Time / Epoch (hr)	1.926	1.844	+0.082	

Table 7: Efficiency comparison between RSPO and DPO. "*Diff.*" denotes efficiency difference, and "*Perf. Diff.*" indicates the performance gain of our RSPO over DPO.

or vanishing gradients, ultimately contributing to improved training stability. A detailed analysis of the gradient dynamics is provided in **Appendix E**.

5.5 Training Efficiency and Performance Analysis

To evaluate the computational overhead of RSPO, we measure average GPU memory usage, training time per epoch, and total memory consumption during training, alongside its performance gains over DPO. As shown in Table 7, RSPO adds a modest 3.19 GB of GPU memory and extends training time by about 4.92 minutes per epoch. Despite this slight increase, RSPO delivers substantial improvements, boosting win rates on AlpacaEval 2 by +4.8% for Mistral-base (7B) and +4.3% for Llama-3-Instruct (8B), evaluated with GPT-4-turbo. These results confirm RSPO as an efficient and effective upgrade over DPO for preference optimization.

6 Conclusion

We propose Reward-Driven Selective Penalization for Preference Alignment Optimization (RSPO), a method that enhances preference alignment by innovatively categorizing data and applying selective weighting. RSPO introduces a dynamic penalty strategy that down-weights noisy or hard-to-align samples during training. Experiments show that RSPO improves alignment and generalization by effectively handling complex preference data.

Limitations

Despite its innovative contributions, the RSPO framework has several limitations, particularly in the design of penalty functions and data partitioning mechanisms.

Impact and Future Exploration of Penalty Functions. Although the proposed heuristic penalty function demonstrates superior performance in the experiments, the choice of penalty function remains a limitation. Different penalty weight functions have varying impacts on the model's training effectiveness, and only a limited set of functions are explored in this study. Future work will investigate a broader range of penalty functions to identify more effective alternatives, with the goal of improving model generalization and training performance. Furthermore, given the diversity and complexity of tasks, it may be necessary to design task-specific penalty functions, suggesting that there is significant potential for further optimization in penalty function design.

Data Partitioning Mechanism also presents challenges. RSPO partitions data into classes based on alignment quality, such as $R1$ (high alignment) and $R2$-$R4$ (lower alignment). Although this classification is theoretically sound, it faces practical challenges. The boundaries between these classes may be ambiguous, and complex data distributions may not fit neatly into predefined categories. This issue is particularly pronounced for intermediate samples that are neither clearly aligned nor completely misaligned. The inefficiency in handling such samples can negatively impact the overall optimization process. Future research should explore more nuanced partitioning strategies that can better accommodate complex data distributions.

References

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, et al. 2021. A general language assistant as a laboratory for alignment. *ArXiv preprint*, abs/2112.00861.

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Rémi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. 2024. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics, 2-4 May 2024, Palau de Congressos, Valencia, Spain*, volume 238 of *Proceedings of Machine Learning Research*, pages 4447–4455.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *ArXiv preprint*, abs/2204.05862.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, volume 382 of *ACM International Conference Proceeding Series*, pages 41–48.

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

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv preprint*, abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *ArXiv preprint*, abs/2110.14168.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2023. Ultrafeedback: Boosting language models with high-quality feedback. *ArXiv preprint*, abs/2310.01377.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. 2024. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *ArXiv preprint*, abs/2404.04475.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. KTO: Model alignment as prospect theoretic optimization. In *International Conference on Machine Learning*.

Duanyu Feng, Bowen Qin, Chen Huang, Zheng Zhang, and Wenqiang Lei. 2024. Towards analyzing and understanding the limitations of dpo: A theoretical perspective. *ArXiv preprint*, abs/2404.04626.

Leilei Gan et al. 2024. A comprehensive survey of direct preference optimization: Datasets, theories, variants, and applications. *ArXiv preprint*, abs/2410.15595.

Raj Gupta, Zaid Alvi, and Priya Sharma. 2022. Improving robustness of language models to noisy annotations. *ArXiv preprint*, abs/2212.09876.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. *ArXiv preprint*, abs/2203.15556.

Jiwoo Hong, Noah Lee, and James Thorne. 2024. ORPO: Monolithic preference optimization without reference model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11170–11189.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations*.

Alice Lee, David Su, and et al. 2023. Aligning language models to complex goals with extensive feedback. *ArXiv preprint*, abs/2301.12345.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Alpacaeval: An automatic evaluator of instruction-following models. `https://github.com/tatsu-lab/alpaca_eval`.

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

Junru Lu, Jiazheng Li, Siyu An, Meng Zhao, Yulan He, Di Yin, and Xing Sun. 2024. Eliminating biased length reliance of direct preference optimization via down-sampled kl divergence. *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1047–1067.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. In *38th Conference on Neural Information Processing Systems (NeurIPS 2024)*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*.

Arka Pal, Deep Karkhanis, Samuel Dooley, Manley Roberts, Siddartha Naidu, and Colin White. 2024. Smaug: Fixing failure modes of preference optimisation with dpo-positive. *ArXiv preprint*, abs/2402.13228.

Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. 2024. Disentangling length from quality in direct preference optimization. In *Findings of the Association for Computational Linguistics: ACL 2024*, page 4998–5017.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Yunhao Tang, Zhaohan Daniel Guo, Zeyu Zheng, Daniele Calandriello, Rémi Munos, Mark Rowland, Pierre Harvey Richemond, Michal Valko, Bernardo Ávila Pires, and Bilal Piot. 2024. Generalized preference optimization: A unified approach to offline alignment. *ArXiv preprint*, abs/2402.05749.

Junkang Wu, Yuexiang Xie, Zhengyi Yang, Jiancan Wu, Jiawei Chen, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. 2024a. Towards robust alignment of language models: Distributionally robustifying direct preference optimization. *ArXiv preprint*, abs/2407.07880.

Junkang Wu, Yuexiang Xie, Zhengyi Yang, Jiancan Wu, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. 2024b. β-dpo: Direct preference optimization with dynamic β. *ArXiv preprint*, abs/2407.08639.

Canwen Xu, Corby Rosset, Ethan C. Chau, Luciano Del Corro, Shweti Mahajan, Julian McAuley, Jennifer Neville, Ahmed Hassan Awadallah, and Nikhil Rao. 2024a. Automatic pair construction for contrastive post-training. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 149–162, Mexico City, Mexico. Association for Computational Linguistics.

Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. 2024b. Contrastive preference optimization: pushing the boundaries of llm performance in machine translation. In *Proceedings of the 41st International Conference on Machine Learning*.

Shusheng Xu, Wei Fu, Jiaxuan Gao, Wenjie Ye, Weilin Liu, Zhiyu Mei, Guangju Wang, Chao Yu, and Yi Wu. 2024c. Is dpo superior to ppo for llm alignment? a comprehensive study. *ArXiv preprint*, abs/2404.10719.

Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. 2023. RRHF: rank responses to align language models with human feedback. In *Advances in Neural Information Processing Systems*.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *ArXiv preprint*, abs/2311.07911.

Wenxuan Zhou, Ravi Agrawal, Shujian Zhang, Sathish Reddy Indurthi, Sanqiang Zhao, Kaiqiang Song, Silei Xu, and Chenguang Zhu. 2024. Wpo: Enhancing rlhf with weighted preference optimization. In *The Conference on Empirical Methods in Natural Language Processing (EMNLP 2024)*.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *ArXiv preprint*, abs/1909.08593.

A RSPO Algorithm

The Reward-Driven Selective Penalization for Preference Alignment Optimization (RSPO) algorithm refines preference optimization by dynamically adjusting the training signal based on implicit reward distributions. Unlike standard Direct Preference Optimization (DPO), which treats all preference pairs equally, RSPO categorizes data into distinct reward scenarios and selectively penalizes samples that may hinder effective preference learning.

Algorithm 1: Reward-Driven Selective Penalization for Preference Alignment Optimization (RSPO)

```
Input: Preference dataset $\mathcal{D}$, policy model $\pi_\theta$,
reference model $\pi_{\text{ref}}$, number of iterations $T$
for $t = 0$ to $T$ do
    Sample a batch of preference pairs $(x, y_w, y_l)$ from $\mathcal{D}$;
    Compute $r(x, y_w)$ and $r(x, y_l)$ for each data point using $\pi_\theta$ and $\pi_{\text{ref}}$;
    if $r(x, y_w) \geq 0$ and $r(x, y_l) \leq 0$ then
        Assign the data to $R1$
    else
        Assign the data to $R2 - R4$
        Assign penalty weights using Equation (2)
    end if
    Compute $L_{\text{RSPO}}$ using Equation (1)
    Update the policy model parameters $\theta$
end for
```

B Details of Multiple Downstream Benchmark Tasks

We present the details of multiple downstream benchmark tasks:

- GSM8K (Cobbe et al., 2021): A generative primary level math dataset of 1.3k questions. We use 8-shot in-context exemplars. We report strict exact match score.

- IFEval (Zhou et al., 2023): A special instruction-following test dataset, contains 541 verifiable instructions. We use 5-shot prompt and report instruction-level strict accuracy.

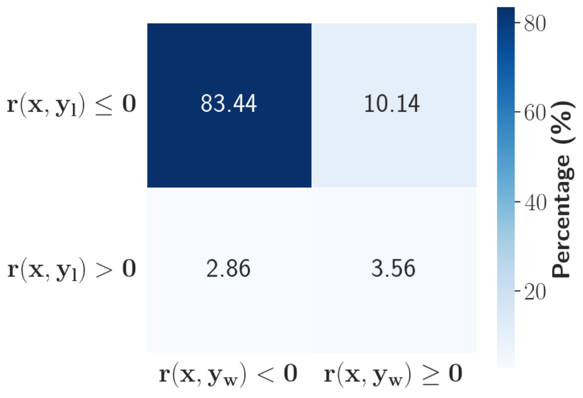
- MMLU (Hendrycks et al., 2021): One of the most popular and largest multi-choice benchmark for testing common knowledge of LLMs, covering 14k questions. We use 5-shot prompt and present accuracy.

- TruthfulQA (Lin et al., 2022): A testing dataset aims for assessing a model's recognition of true statements. We evaluate all 817 questions with 0-shot prompt,and reporting truthfulqa_mc1 accuracy score.

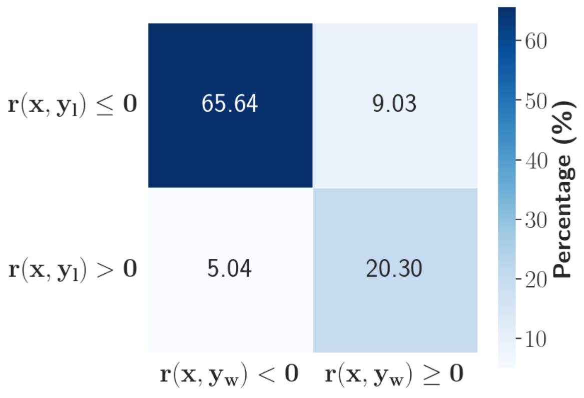
- ARC (Clark et al., 2018): A multiple-choice benchmark for science questions from grades 3 to 9, split into Easy and Challenge parts. The Challenge part has harder, reasoning-based questions. We evaluate all 817 questions with 25-shot prompt, and reporting accuracy score.

C Experiment Parameters

Based on the DPO parameters provided by the Princeton-NLP team, we achieved significant improvements by solely adjusting the penalty weight coefficients. Additionally, we set the max_length for the Llama-3-Instruct model to only half of what the Princeton-NLP team set, which reduces training time by nearly 50%. Table 8 provides detailed experimental parameters for Mistral-base and Llama-3-Instruct.



(a) Mistral-SFT.



(b) Llama-3-Instruct.

Figure 6: The proportion of the four types of preference response pairs during the DPO training process for Mistral-SFT and Llama-3-Instruct.

Parameter	**Mistral-base**	**Llama-3-Instruct**
GPU	8×Ascend910B	8×Ascend910B
beta	0.01	0.01
batch	128	128
learning_rate	5e-7	7e-7
max_prompt_length	512	512
max_length	1024	1024
num_train_epochs	1	1
torch_dtype	bfloat16	bfloat16
warmup_ratio	0.1	0.1
β_w	0.01	0.01
β_l	0.1	0.1
λ	0.1	0.1

Table 8: Experimental Parameters for Mistral-base and Llama-3-Instruct.

D Proportion of the Four Types of Preference Response Pairs

Figure 6 shows the distribution of preference response pairs during the DPO training process for Mistral-Base and Llama-3-Instruct. As seen in Figure 6, the preference response pairs for Mistral-Base are primarily concentrated in the negative preference category, especially in the $r(x, y_l) \leq 0$ and $r(x, y_w) < 0$ category, which accounts for 83.44% of the cases, indicating that the model tends to predict negative preferences. In contrast, the preference response distribution for Llama-3-Instruct is more balanced. While the largest proportion is still concentrated in the negative preference category, the distribution across the other categories is relatively more spread out.

It is necessary to distinguish between these four categories because they represent different types of model behavior, each of which could impact the training process and final performance. The cases where both responses are negative ($r(x, y_l) \leq 0$ and $r(x, y_w) < 0$) suggest that the model may be overly conservative in assigning preference to responses. On the other hand, when $r(x, y_l) \leq 0$ and $r(x, y_w) \geq 0$, it indicates that the model is making a more confident prediction that contradicts the actual preference, suggesting a misalignment that could be problematic for generalization.

Differentiating between these categories helps ensure that the model is not overfitting to one type of preference signal (e.g., negative preferences) while neglecting others. Furthermore, it allows for targeted interventions, such as adjusting the training weights for different categories based on their quality, complexity, or relevance, improving the model's ability to generalize and handle ambiguous or noisy preference data effectively. This is especially critical when optimizing performance in a real-world setting where preferences may not always be clearly defined.

E Gradient Analysis

Previous studies have shown that DPO exhibits significant asymmetry in the gradient signals between chosen and rejected responses during training, specifically manifested as:$|\nabla_\theta \log \pi_\theta(y_i|x)| > |\nabla_\theta \log \pi_\theta(y_{v_i}|x)|$.

This asymmetry causes the model to rapidly decrease the generation probability of rejected responses while failing to effectively increase the probability of chosen responses. The imbalance in DPO's learning process—where the gradient weight for chosen responses is smaller than that for rejected responses—makes it difficult for the model to learn chosen responses effectively.

λ	β_w	β_l	**ARMO WR**
0.1	0.01	0.1	27.20%
0.0	0.01	0.1	17.01%
0.1	0.1	0.01	20.49%
0.1	0.01	0.01	23.72%
0.1	0.1	0.1	20.37%

Table 9: Hyperparameter exploration results for the proposed RSPO method.

Functions	**Objective**
f_1	$\lambda + \sigma\left(\beta_w \log \frac{\pi_\theta(y_w\|x)}{\pi_{\text{ref}}(y_w\|x)}\right)$
f_2	$\lambda \cdot \sigma\left(\beta_w \log \frac{\pi_\theta(y_w\|x)}{\pi_{\text{ref}}(y_w\|x)}\right) + \sigma\left(\beta_l \log \frac{\pi_{\text{ref}}(y_l\|x)}{\pi_\theta(y_l\|x)}\right)$
f_3	$\lambda + \sigma\left(\beta_w \log \frac{\pi_\theta(y_w\|x)}{\pi_{\text{ref}}(y_w\|x)}\right) \cdot \sigma\left(\beta_l \log \frac{\pi_{\text{ref}}(y_l\|x)}{\pi_\theta(y_l\|x)}\right)$

Table 10: Different penalty weight functions.

According to curriculum learning theory, forcing the model to fit difficult problems too early can lead to suboptimal optimization. We believe that in this scenario, assigning a higher weight to $R1$-type data can stabilize the learning process and achieve better optimization performance.

We compared the changes in unclipped gradients between our method and DPO during training. We observed that DPO exhibited greater gradient fluctuations, with a sharp increase between steps 50 and 75, followed by a consistently high level.

This occurs because, during training, the model develops some understanding of the training data, but the data remains more complex than what the model can fully comprehend. As a result, the model struggles to capture the information in the samples completely. In this scenario, the model undergoes large gradient updates continuously, indicating that DPO experiences highly unstable gradient updates throughout the training process.

In contrast, our method (RSPO) assigns higher weights to $R1$-type data, which aligns with the model's gradient update direction. This reduces the impact of hard-to-learn data on gradient updates, leading to more consistent updates during training. As a result, the optimization process becomes more stable, exhibiting a smoother gradient trajectory with smaller fluctuations.

F Penalty Function Design and Hyperparameter Exploration

We further supplement our analysis of the penalty function design and the corresponding hyperparameter exploration as follows:

First, based on the experimental results presented in Figure 3 and Table 5, we observed that assigning a higher weight to $R1$-type data can potentially improve the model's overall performance. Motivated by this observation, we considered increasing the weight assigned to $R1$ samples by proportionally reducing the weight of other sample types.

Our initial approach was to introduce a constant as a penalty function. However, considering that Direct Preference Optimization (DPO) is inherently a pairwise preference learning method, simply using a constant could achieve some effect but would neglect the relationship between paired preferences. Specifically, previous studies have shown that the model tends to more easily suppress the probability of rejected responses than to enhance the probability of chosen responses. Hence, the model's ability to increase $\pi_\theta(y_w|x)$ and decrease $\pi_\theta(y_l|x)$ is asymmetric.

To address this, we propose a dynamic penalty function that separately considers the chosen and rejected responses (Table 10). Drawing inspiration from the implicit reward formulation in DPO,

$$r(x, y) = \beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}$$

we design our penalty function as:

$$f(y_w, y_l, x) = \lambda + \sigma\left(r(x, y_w)\right) \cdot \sigma\left(-r(x, y_l)\right)$$

that is,

$$= \lambda + \sigma\left(\beta_w \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)}\right) \cdot \sigma\left(\beta_l \log \frac{\pi_{\text{ref}}(y_l|x)}{\pi_\theta(y_l|x)}\right)$$

where $\sigma(\cdot)$ denotes the Sigmoid function.

Here, the term $\log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)}$ captures the policy model's relative ability to increase the probability of generating the *chosen* response compared to the reference model. A higher value implies better generation quality, which results in a larger weight after passing through a scaled Sigmoid transformation parameterized by β_w.

Similarly, $\log \frac{\pi_{\text{ref}}(y_l|x)}{\pi_\theta(y_l|x)}$ reflects the model's ability to suppress the *rejected* response. If the policy model assigns a high probability to a rejected response, a lower weight is accordingly applied after

Model	Method	LC	WR	Std. Error
Mistral-sft	DPO	20.6	18.2	1.19
Mistral-sft	RSPO	25.4	23.7	1.25
Llama-3-Instruct	DPO	40.7	38.1	1.52
Llama-3-Instruct	RSPO	45.0	25.2	1.46

Table 11: Main results with standard errors (3 runs, median reported).

Task	Accuracy	Standard Error
GSM8K	37.45	0.01245
ARC	57.94	0.01446
TQA	47.25	0.01747
MMLU	58.58	0.00395
IFEVAL	55.04	0.02131

Table 12: Standard errors of RSPO on downstream tasks.

mapping through the Sigmoid function scaled by β_l.

Given that models more easily learn to suppress rejected responses than to promote chosen ones, we set different scaling factors: specifically, $\beta_l = 0.1$ to amplify the sensitivity to the suppression of rejected responses, and $\beta_w = 0.01$ to attenuate the sensitivity when promoting chosen responses.

Additionally, we introduce a constant term λ to ensure a minimum gradient contribution from each data point, preventing the dynamic penalty from becoming excessively small, which could otherwise impede effective learning on certain examples.

We further conducted hyperparameter exploration for the proposed penalty function. The experimental results are summarized in Table 9.

Through these ablation studies, we conclude that assigning a relatively larger value to β_l compared to β_w, combined with introducing a constant term λ, leads to better model performance and more stable optimization. The experimental results also validate our initial motivation in designing the penalty function: namely, to encourage the model to focus on learning interpretable content while maintaining a non-negligible learning signal for each sample.

G Evaluation Stability and Standard Error Reporting

To provide a more comprehensive evaluation of our method, we conducted multiple experimental runs and report standard errors alongside key results. For both DPO and our proposed RSPO method, we performed three independent training runs and report the median results, along with their standard errors. These are presented in Table 11 for different models and evaluation settings. For other baseline methods such as SimPO and WPO, we cite results directly from their respective papers, which involved extensive hyperparameter tuning and repeated testing to ensure reliability.

Furthermore, we also evaluated the stability of our method on downstream tasks. While prior works often omit reporting standard deviations for such tasks, we include the standard errors of accuracy across multiple runs in Table 12. These results help quantify the robustness of our method across different tasks, demonstrating that RSPO achieves consistent performance with low variance.