

MWPO: Enhancing LLMs Performance through Multi-Weight Preference Strength and Length Optimization

Shiyue Xu, Fu Zhang*, Jingwei Cheng, Linfeng Zhou

School of Computer Science and Engineering, Northeastern University, China
zhangfu@mail.neu.edu.cn, xu_2001@foxmail.com

Abstract

Direct Preference Optimization (DPO) have proposed offline alternatives to Reinforcement Learning from Human Feedback (RLHF). In DPO, each preference pair, which serves as the foundation for learning, is typically constructed by first generating multiple responses to the same instruction and then annotating them to indicate the preferred choice. However, when the responses are highly similar, the weak preference signal can introduce annotation noise, which may hinder model optimization. Additionally, DPO suffers from the drawback of over-optimizing for verbose generation. A potential reason is the presence of length bias in preference datasets, which can lead to length exploitation. To address these issues, we propose a DPO-based **multi-weight preference strength and length optimization (MWPO)** method. Specifically, we propose to reweight preference pairs based on implicit reward margins and response length margins, unifying them through a geometric mixture to generate synthetic weights for optimization. This method allows preference pairs with stronger preference signals or more favorable length feature to have a more pronounced impact on model parameters. Moreover, our method does not require additional annotators. We validate our method on models of four different scales across multiple benchmarks. Our method surpasses state-of-the-art (SOTA) baselines, outperforming DPO by up to 8.7% on AlpacaEval 2 while reducing generation length by 9.4% in the Mistral setting¹.

1 Introduction

Reinforcement Learning from Human Feedback (RLHF) is a key approach for effectively aligning large language models (LLMs) with human values

and preferences (Ouyang et al., 2022; Ziegler et al., 2019; Bai et al., 2022). RLHF further improves the instruction-following capability and safety of supervised fine-tuned (SFT) models. In traditional RLHF pipelines, the reinforcement methods such as Proximal Policy Optimization (PPO) (Schulman et al., 2017), have demonstrated their effectiveness in popular LLM applications like ChatGPT (Achiam et al., 2023) and Claude (Anthropic, 2024). However, due to its multi-stage online process, which involves training reward models and optimizing policy models, RLHF presents significant challenges in training.

Recently, offline preference optimization algorithms, represented by methods like Direct Preference Optimization (DPO) (Rafailov et al., 2024c), have emerged as highly promising approaches. DPO reparameterizes the reward function used in RLHF, eliminating the need for a reward model and directly learning the policy model from offline preference data. Due to its simplicity and efficiency, DPO has been widely adopted in practice.

In offline preference optimization methods, data quality directly determines alignment performance. Therefore, making efficient use of existing offline data is crucial. DPO and its variants, such as SimPO (Meng et al., 2024), *treat each preference pair equally*, making them vulnerable to certain noise in datasets. Specifically, each preference pair in an offline dataset is typically constructed by annotating preferences for multiple responses sampled from the same prompt. Yang et al. (2024) and Wang et al. (2024a) also argue that using the same prompt to construct preference pairs often results in responses of similar quality, making it challenging to distinguish between them. In this scenario, the data with narrow preference margins can lead to training signals being overshadowed by noisy preference labels, ultimately impairing the performance of the trained model. This inspired us to consider that *distinct preference pairs might be*

* Corresponding author.

¹Our code is available at <https://github.com/AIR-h1/MWPO>.

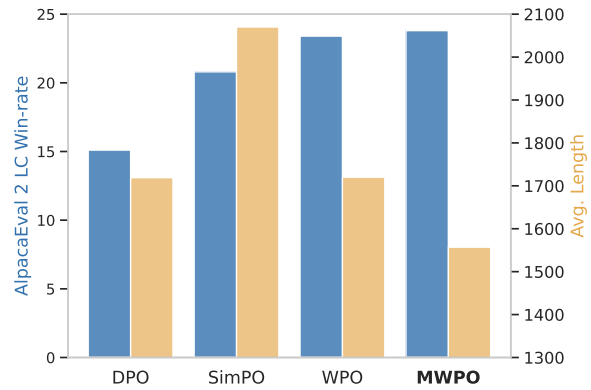
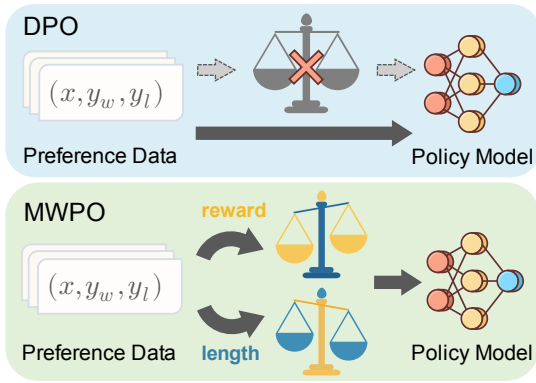


Figure 1: Overview of our Multi-Weight Preference strength and length Optimization (MWPO) and results on AlpacaEval 2. Vanilla DPO treats each preference data equally, while MWPO optimizes the preference strength and length by weighting the data from different aspects. SimPO (Meng et al., 2024) and WPO (Zhou et al., 2024) are recent SOTA baselines.

more beneficial for model optimization. Therefore, it may be valuable to explore how to effectively distinguish and leverage each preference pair in offline data for model optimization.

Additionally, another significant challenge faced by preference optimization methods is "verbosity", where the optimized models tend to generate much longer responses without necessarily improving the generation quality. Analysis shows that policy models trained with DPO generate responses that are significantly longer than the base model (Park et al., 2024). This problem is related to bias in the length distribution of the training data, where annotators typically prefer more detailed responses, causing the chosen responses to be longer than the rejected responses. Models may exploit this bias, leading to a statistically significant increase in response length.

To address these issues, in this paper, we propose a DPO-based **multi-weight preference strength and length optimization (MWPO)** method. As shown in Fig. 1, unlike DPO which treats each data equally, our method measures the preference data from two aspects. *First*, to effectively distinguish and leverage each preference pair, we argue that during training, the policy model can act as a preference discriminator leveraging previously learned preference information to assess preference differences in new data. Strongly preferred data are given more attention during training. To achieve this, we propose a **reward-margin weight**. *Second*, to address the potential length bias issue, we construct an effective **length-controlled weight** based on the scaled length margin between the chosen and re-

jected responses. *Finally*, we can integrate both or one of the weights with the loss of DPO to obtain the final training objective. Our contributions are summarized as follows:

- We propose a preference-weighting method that leverages implicit rewards from the policy model itself during training, without additional reward models. The preference pairs with higher preference strength will have a greater impact on model parameters.
- We propose a length-controlled method by weighting the preference data based on the scaled length margin between the chosen and rejected responses. Our length-controlled weight can effectively mitigate length exploitation and flexibly adjust the constraint strength through coefficients.
- We use four LLM models of different architectures and parameter scales to validate the effectiveness of our method, achieving better performance than DPO and recent SOTA baselines on multiple benchmarks, and significantly reducing the average generation length.

2 Related Work

RLHF has become an essential method to align or enhance the performance of LLMs (Ouyang et al., 2022; Touvron et al., 2023; Dubey et al., 2024; Young et al., 2024; Jiang et al., 2024; Li et al., 2023b). RCLD (Yang et al., 2024) employs contrasting instructions to create differentiated preference data. West-of-N (Pace et al., 2024) samples multiple responses and selects the best and

the worst response through an additional reward model to construct a preference pair. However, reinforcement learning methods still suffer from data and training inefficiency due to the need for repeated sampling and policy updates during training (Rafailov et al., 2024c; Tang et al., 2023).

More offline methods, exemplified by DPO (Rafailov et al., 2024c), typically optimize policy models using offline preference dataset without additional reward model. GPO (Tang et al., 2023) unifies several offline methods, including DPO, IPO (Azar et al., 2024), and SliC (Zhao et al., 2023). KTO (Ethayarajh et al., 2024) learns from a binary signal of whether an output is (un)desirable. WPO (Zhou et al., 2024) simulates the on-policy learning process by adding weights to the optimization objective of DPO. None of these methods account for the potential noise introduced by similar responses.

Another significant challenge in reinforcement learning is over-optimization, where reward scores from the reward model increase, but the quality of model generation deteriorates, often becoming excessively verbose (Zheng et al., 2023; Pitis, 2023; Rafailov et al., 2024a; Lu et al., 2024). IPO (Azar et al., 2024) mitigates the risk of DPO over-optimization through theoretical analysis. Meng et al. (2024) propose SimPO, which avoids reliance on a reference model by using length-normalized rewards to prevent verbose responses. R-DPO proposed by Park et al. (2024), effectively controls the response length by simply adding a length regularization term to the implicit reward.

In this work, we extend the DPO method by jointly optimizing the preference strength and length from different aspects. Unlike existing offline methods, we reweight each preference pair according to its preference strength. Additionally, we introduce a length-controlled weight to replace the additive length regularization, which effectively mitigates length exploitation and allows for flexible adjustment of the constraint strength through coefficients. We benchmark our method against a series of offline preference optimization methods, including DPO, SimPO, WPO, and et al. Results demonstrate that our method achieves a good performance and controls the generation length effectively.

3 Method

In this section, we first present the theoretical background of RLHF and DPO (§3.1). Then, we propose two distinct weights: one for measuring pref-

erence strength (§3.2) and the other for addressing length bias (§3.3). Finally, we construct a mixed weight approach (§3.4) and provide a theoretical analysis of our method (§3.5).

3.1 Preliminary

DPO (Rafailov et al., 2024c) unifies reward modeling and reinforcement learning fine-tuning of RLHF into a single-stage optimization through theoretical derivation. Specifically, DPO reparameterizes the reward model (RM) r_ϕ as the implicit reward function follows:

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

where π_θ and π_{ref} are policy model and reference model respectively. π_θ and π_{ref} are usually initialized from same SFT model; $Z(x)$ is the partition function that can be eliminated in contrastive loss; β is a hyperparameter that adjust the constraint strength. By maximizing the reward margin between the chosen response y_w and the rejected response y_l based on the Bradley-Terry (B-T) (Bradley and Terry, 1952) model, the DPO loss is obtained as follows:

$$\mathcal{L}_{DPO}(\pi_\theta; \pi_{ref}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} [\log \sigma(r_w - r_l)]$$

where r_w and r_l denote the implicit reward of y_w and y_l , respectively; σ stands for sigmoid function.

R-DPO (Park et al., 2024) introduces a length regularization term $\alpha|y|$ into the implicit reward function:

$$r(x, y) = \beta \log \frac{\pi_\theta(y|x)}{\pi_{ref}(y|x)} + Z(x) - \alpha|y|$$

where $|y|$ is the token-length of the response y , then gets the regularized-DPO objective:

$$\mathcal{L}_{R-DPO}(\pi_\theta; \pi_{ref}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} [\log \sigma(r_w - r_l - \alpha(|y_w| - |y_l|))]$$

3.2 Reward-margin Weight

The training of DPO essentially constitutes a reward modeling process utilizing the B-T model, thereby we assume that the model itself can provide reference information on the preference differences in the training data. This judgment is initially based on the model’s prior knowledge and is continuously reinforced throughout training. Specifically, during the training process, the implicit rewards calculated by the model for the chosen and rejected responses are used to construct a weighted judgment

on the preference difference between a pair of data. To this end, we propose a *reward-margin weight* method to measure preference strength, which can effectively distinguish and utilize each preference pair in offline data for model optimization.

$$w_{rw}(y_w, y_l, x) = \sigma(r_w - r_l) \quad (1)$$

Furthermore, the implicit reward equation, excluding the partition function $Z(x)$, can be reformulated at the token-level from its original sequence-level representation, as Eq.2 (the detailed derivation is provided in Appendix A).

$$r(x, y) = \beta \sum_{t=1}^{|y|} \log \frac{\pi_{\theta}(y^{(t)}|x; y^{<t})}{\pi_{ref}(y^{(t)}|x; y^{<t})} \quad (2)$$

where $|y|$ denotes the length of sequence y , $y^{(t)}$ denotes the t -th token and $y^{<t}$ denotes all the tokens before the t -th token in sequence y . According to the Theorem 1 in Rafailov et al. (2024b), DPO can learn the optimal strategy for any per-token reward function; therefore, the sequence-level implicit reward in the contrastive loss can be viewed as the cumulative sum of token-level implicit rewards. It is intuitive that the calculation of implicit rewards introduces a strength bias. Even if two responses have similar single-token reward, the cumulative calculation leads to a higher total reward for longer responses, regardless of whether their ground-truth reward is actually higher. To mitigate the aforementioned bias, we introduce a second weight to control for length, with specific details to be explained in §3.3.

3.3 Length-controlled Weight

As mentioned in §3.1, R-DPO (Park et al., 2024) introduces a length regularization term in the implicit rewards of DPO to impose constraints. However, in practice, we find that due to possible significant variations in reward values across different datasets, policy models and hyperparameters settings, incorporating the length control term additively may result in optimization challenges. Therefore, we propose to address the potential length bias issue by constructing a simple *length-controlled weight* based on the scaled length margin between the chosen and rejected responses. We introduce a length-controlled term $\lambda_{lc} \cdot (|y_l| - |y_w|)$, which is used as a weight to the data through the sigmoid function:

$$w_{lc} = \sigma(\lambda_{lc} \cdot (|y_l| - |y_w|)) \quad (3)$$

where λ_{lc} is a scaling factor. As the length of the chosen response exceeds that of the rejected response, the length-controlled weight decreases, diminishing the impact of these preference pairs.

3.4 Mixed Weight

For multiple weights, we can apply a geometric mixture approach to unify Eq.1 and Eq.3 into a single mixed weight, as shown in Eq.4. Intuitively, we prefer the data with a certain degree of preference strength and minimal length bias.

$$w_{mixed} = w_{rw}^{\alpha} \cdot w_{lc}^{(1-\alpha)} \quad (4)$$

By adjusting the mixing coefficient α , the influence of different weights on the mixed weight can be altered. Theoretically, the sum of the control coefficients for w_{rw} and w_{lc} need not equal 1, but for simplicity in subsequent experiments, we only consider the case where their sum equals 1. Combining this mixed weight with the DPO loss yields the final training objective, as follows:

$$\mathcal{L}_{MWPO} = w_{mixed} \cdot \mathcal{L}_{DPO} \quad (5)$$

3.5 Theoretical Analysis of MWPO

To further understand the impact of the constructed weights on training, a common approach is to analyze the gradients of the loss function to uncover the underlying mechanisms. The gradient of the DPO loss function with respect to the parameters θ is given by:

$$\nabla_{\theta} \mathcal{L}_{DPO} = -\beta \mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\sigma(r_l - r_w) \cdot \left(\frac{1}{\pi(y_w)} \nabla_{\theta} \pi(y_w) - \frac{1}{\pi(y_l)} \nabla_{\theta} \pi(y_l) \right) \right] \quad (6)$$

Since the weights we constructed do not need backpropagation in training, the gradient for MWPO differs from DPO only by the weight term, $w_{DPO} = \sigma(r_l - r_w)$. First, we focus on the combined weight term consist of $w_{rw}^{\alpha} \cdot w_{DPO}$ as Eq.7:

$$w_{rw}^{\alpha} \cdot w_{DPO} = \sigma(r_w - r_l)^{\alpha} \cdot \sigma(r_l - r_w) \quad (7)$$

In the raw gradient formula of DPO, the greater the difference between $r_l - r_w$, the higher the weight assigned, meaning a punishment will be given when the reward estimate is wrong. However, our w_{rw} behaves oppositely, assigning a higher weight when the reward estimate is correct. ***Does this cause the model to focus solely on simple data?*** The answer is "No". The combined weight

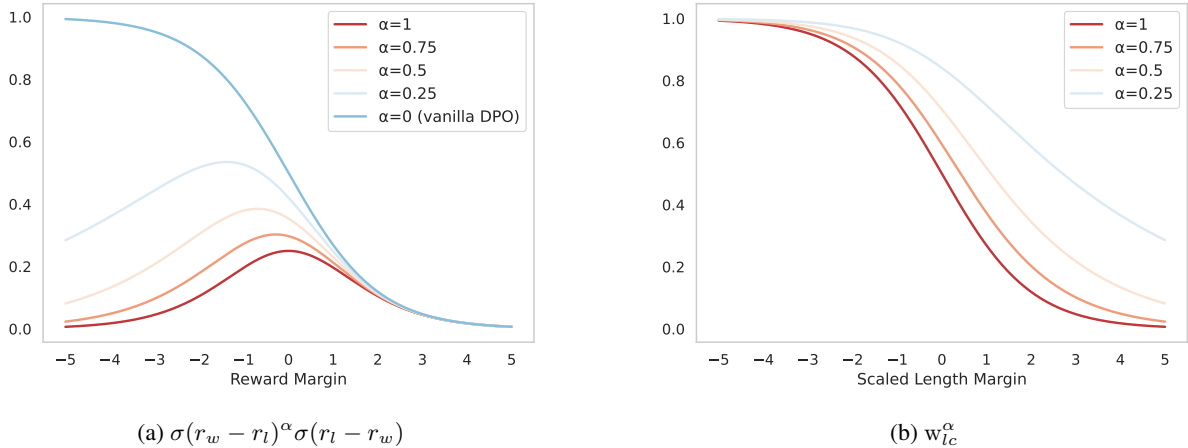


Figure 2: Function plots for weights under different constraint strengths. (a) Combination of reward-margin weight and the weight in the gradient of DPO. (b) Length-controlled weight with different control strengths.

term for different values of α is shown in Fig. 2a. It is a unimodal function with a negative extremum. Within a certain range of values, it does not change the fundamental properties of the DPO gradient weight, but rather reduces the weight when the estimated bias is too large. We suggest that, if the rejected response has higher model confidence or the chosen response has lower confidence based on previously learned preferences, reducing the weight to update parameters cautiously helps prevent potential noise from interfering with the current preference distribution. We will conduct more in-depth analysis in subsequent experiments.

4 Experimental Setup

In this section, we begin with the models and training settings (§4.1); followed by the evaluation benchmarks (§4.2) and baselines (§4.3).

4.1 Models and Training Settings

Due to the potentially significant differences in behavior across models, we select four models with varying parameter scales to ensure the robustness of the results. These models include Mistral-7B-Base (Jiang et al., 2023a), Llama3-8B-Instruct (AI@Meta, 2024a), Llama3.2-3B-Base (AI@Meta, 2024b) and Qwen2.5-1.5B-Base (Team, 2024).

For all base models without instruction fine-tuning, we follow zephyr (Tunstall et al., 2023) and fine-tune them on Ultrachat_200k (Ding et al., 2023; Tunstall et al., 2023) to obtain the SFT models. We perform preference optimization on ultrafeedback-binarized² (Tunstall et al., 2023),

²https://huggingface.co/datasets/HuggingFaceH4/ultrafeedback_binarized

which contains 64k preference pairs derived from the original Ultrafeedback dataset (Cui et al., 2024). For simplicity, we refer to it as Ultrafeedback throughout the paper.

For the instruction-tuned models, we follow Meng et al. (2024), using off-the-shelf instruction-tuned models as the SFT models and utilizing the sampled preference data³ annotated by PairRM (Jiang et al., 2023b) to perform training.

We perform full-parameter fine-tuning for all models. To minimize the impact of hyperparameters, we use the Mistral weights for the baselines, which are trained with full parameters and open-sourced by the princeton-nlp group following an extensive hyperparameter search. For the other models, we conduct a thorough hyperparameter search to ensure a fair comparison. Detailed training and evaluation settings are provided in Appendix B.

4.2 Evaluation Benchmarks

For the Mistral setup, we use AlpacaEval 2 (Li et al., 2023a) which is a popular opened general automated metric that measures LLMs’ alignment with human preferences using 805 representative instructions. For each instruction, we follow the default evaluation setting that using GPT-4-1106-preview as the annotator. We report the length-controlled (LC) win-rates (Dubois et al., 2024) against GPT-4 and the average output length. LC win-rates are a debiased version of the win-rates that control for the length of the outputs.

Additionally, we also make a more evaluation of each method on benchmarks for multiple down-

³<https://huggingface.co/datasets/princeton-nlp/llama3-ultrafeedback>

stream tasks, including: GSM8K (Cobbe et al., 2021), MMLU (Hendrycks et al., 2021), ARC (Clark et al., 2018), TruthfulQA (Lin et al., 2021), GPQA (Rein et al., 2023). We report the strictly match accuracy of conditional benchmarks.

For the other models, since the high cost of using GPT-4 for evaluation, we choose DeepSeek-v3 as the proxy annotator and report the LC win-rate on AlpacaEval 2. We evaluate nine different models using DeepSeek and find that the results exhibit a high correlation with the official GPT-4 evaluation, demonstrating the effectiveness of the proxy model. Detailed results are provided in Appendix C.

4.3 Baselines

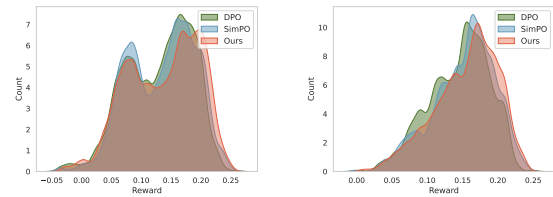
We compare our method against recent state-of-the-art offline preference optimization methods. DPO (Rafailov et al., 2024c) is the most representative pairwise preference optimization method which is without reinforcement learning; RRHF (Yuan et al., 2024) uses probabilities rank loss and includes an SFT objective; IPO (Azar et al., 2024) is a very theoretically method that analyzes the possible risk of over-optimization in DPO and avoids DPO’s assumptions by using a squared loss; KTO (Ethayarajh et al., 2024) uses economic theory and can learn from unpaired preference data; CPO (Xu et al., 2024) constructs a contrastive sequence likelihood loss without reference model through formula derivation; R-DPO (Park et al., 2024) introduces an additional regularization term to prevent length exploitation; WPO (Zhou et al., 2024) adapts off-policy data to resemble on-policy data more closely by reweighting preference pairs according to their current probability. SimPO (Meng et al., 2024) uses average probability to eliminate length reliance and eliminate the reference model.

5 Experimental Results

In this section, we first report the group comparison experiments (§5.1) against strong baselines on a series benchmarks, our methods make a superior performance and length control. Then, we study the two weights in Section 5.2 and 5.3 respectively.

5.1 Main Results

The main results in Table 1 show that *our method exhibits superior overall performance* in the Mistral setting. On AlpacaEval 2, our MWPO achieves LC win-rate of 23.8%, significantly outperforming most baselines. Notably, compared to DPO, our



(a) rewards on Ultrafeedback (b) rewards on AlpacaEval

Figure 3: The rewards distributions of the methods as evaluated by Armo on Ultrafeedback and AlpacaEval.

model demonstrates 8.7% improvement in LC win-rate, while also outperforming DPO in generation length. Additionally, MWPO outperforms recent SOTA methods SimPO and WPO in LC win-rate and achieves a 9.4% reduction in average generation length. We also report the win-rate results of different mixture methods in Appendix D.1.

On the five downstream task benchmarks, we also achieve notable improvements over DPO, with average strict match accuracy of 1.2%, particularly in ARC (4%), and TruthfulQA (4.4%). Compared with all baselines, *our model achieves the highest average accuracy*. The overall experimental results demonstrate that our model exhibits superior comprehensive performance across multiple metrics, while achieving a good balance between performance and response length.

Table 2 presents the LC win-rates of the preference optimized models compared to GPT-4 and the SFT model on AlpacaEval 2, as evaluated by DeepSeek-v3, along with the average generation length. *We achieve the highest LC win-rate against GPT-4 across all setups and the highest win-rate against SFT*, except for the Llama-3-8B-instruct setup. The results for GPT-4 indicate that our method generates more high-reward responses, while the results for SFT show that our method produces a number of high-reward responses overall. In terms of length control, our method also exhibits stability, consistently outperforming DPO.

In addition, we further report the inference performance of the model under the Mistral setup across different datasets in Fig. 3. We use the proxy reward model ArmoRM-Llama3-8B-v0.1 (Wang et al., 2024b) to obtain the rewards distributions of generations on Ultrafeedback-binarized test data (Fig. 3a) and AlpacaEval dataset (Fig. 3b), respectively. It is evident that our method results in responses with a higher rewards across both datasets than DPO.

Mistral-7B-Base								
Method	AlpacaEval 2.0		GSM8K	ARC-C	TruthfulQA	MMLU	GPQA	Average
	LC	Avg. Len						
SFT	8.4	914	42.61	55.97	28.27	57.17	29.69	42.74
DPO	15.1	1719	33.13	59.64	46.14	57.46	28.12	44.90
RRHF	11.6	1463	38.29	50.17	32.31	51.55	28.55	40.17
IPO	13.1	1380	36.01	55.89	38.19	58.55	27.23	43.17
KTO	14.4	1144	41.39	60.17	41.01	59.04	25.67	45.55
CPO	8.9	1827	34.57	51.19	36.59	56.45	32.14	42.18
R-DPO	17.4	1335	30.10	56.06	40.64	58.48	27.23	42.50
SimPO	20.8	2070	33.59	60.15	43.45	58.25	27.47	44.58
WPO	23.5	1720	31.31	61.26	40.51	58.53	25.44	43.34
Ours	23.8	1557	30.01	63.66	50.53	57.37	29.24	46.16

Table 1: Main results of the Mistral settings on benchmark datasets and AlpacaEval 2. All models, have undergone extensive hyperparameter searching. The *Avg. Len* refers to the average number of output tokens for AlpacaEval.

Method	Qwen2.5-1.5B-Base			Llama3.2-3B-Base			Llama3-8B-Instruct		
	vs GPT-4	vs SFT	Avg. Len	vs GPT-4	vs SFT	Avg. Len	vs GPT-4	vs SFT	Avg. Len
SFT	4.81	-	1106	6.21	-	1075	29.76	-	1920
DPO	11.17	76.28	1871	11.60	75.91	1960	43.12	73.07	1883
R-DPO	9.90	77.15	1335	10.41	75.18	1815	47.44	75.03	1893
SimPO	10.95	75.75	2154	7.64	49.80	1038	46.30	80.19	1870
WPO	12.42	73.87	2005	11.42	75.08	1896	33.04	58.59	1840
Ours	12.87	81.85	1656	14.89	76.81	1816	49.48	74.97	1703

Table 2: LC win-rate evaluated by DeepSeek-v3 on AlpacaEval 2, across models with varying architectures and parameter scales.

5.2 Study on Reward-margin Weight

Reward-margin weight improves the performance but leads to longer generation. We train Mistral with only reward-margin weight and inference on AlpacaEval dataset, then compute the LC win-rates against GPT-4 generation judged by DeepSeek-v3, the result is showed in Fig. 4a. Due to the resource constraints, we only conducted experiments with LoRA. The LC win-rates are consistently higher than those of DPO, but at the same time, as the control strength (α) increases, the average response length also increases significantly, and exceeds that of the DPO generation consistently. This indicates that, despite the increase in response length, the diversity and detail of the DPO responses also improve.

Reason analysis for longer generation with only reward-margin weight. Our experiments and analysis reveal that the positive correlation between the reward margin and the length margin results in longer generations. As discussed in Section 3.2 regarding Eq.2, the cumulative calculation of im-

PLICIT reward might make longer responses more likely to attain higher reward values under same conditions. And, training on a dataset with positive length bias, where chosen responses are generally longer than rejected responses, is also more likely to cause the model to assign higher token rewards to a longer response.

The results in Fig. 5d also shows that the implicit reward margin and length margin are positively correlated under current settings. Consequently, preference pairs with larger length margins tend to carry higher reward-margin weights, thereby exerting a greater impact on model parameters.

5.3 Study on Length-controlled Weight

Length-controlled weight can control the generation length effectively. We train Mistral with only the length-controlled weight and other settings are the same as in Section 5.2, the result is shown in Fig. 4b. As expected, the average response length decreases with increasing control strength, consistently remaining more than 20% shorter than that

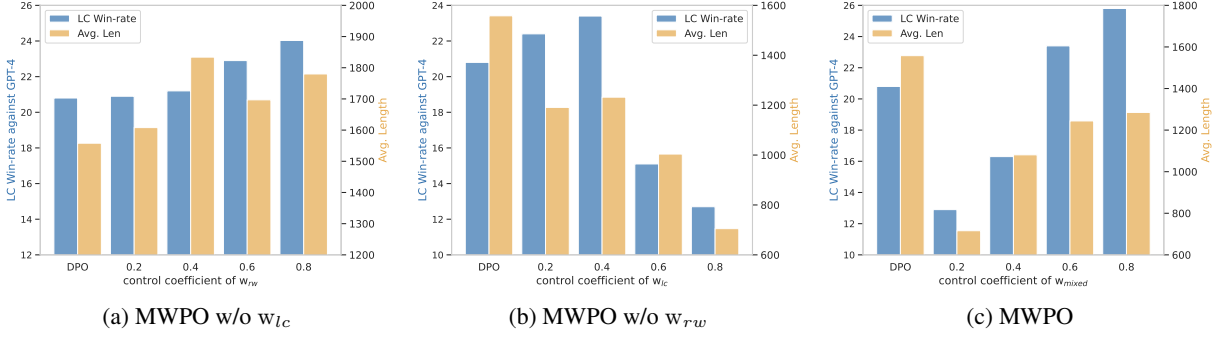


Figure 4: Ablation results for different weights with different control coefficient on AlpacaEval dataset. (a) Performance of DPO with only reward-margin weight. (b) Performance of DPO with only length-controlled weight. (c) Performance with complete mixed weight (MWPO).

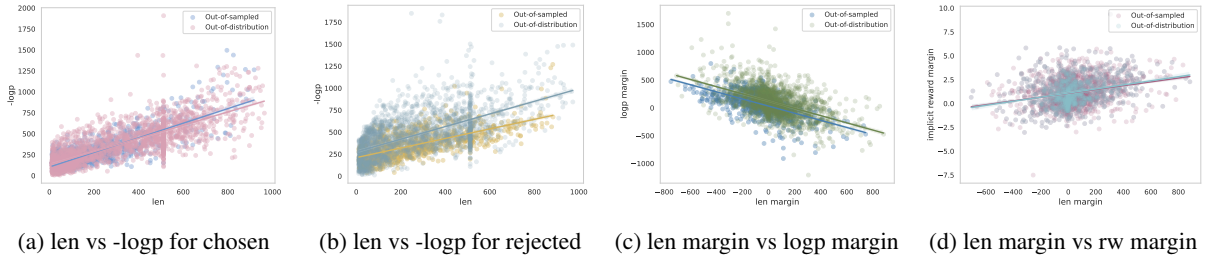


Figure 5: The distribution statistics of preference data when training DPO. We use a trained DPO checkpoint and perform statistics both on the out-of-sampled data and out-of-distribution data. It should be noted that (a) and (b) illustrate the relationship between negative log probability and length. *rw margin* denotes implicit reward margin.

of DPO. LC win rates are higher than DPO when the control strength is moderate, but if the control strength is too high, it may lead to overly brief responses that omit key information. We report the results on the influence of another hyperparameter, λ_{lc} , in Appendix D.2.

By eliminating the common components between the chosen response and rejected response in Eq.6, we can deduce Eq.8, it becomes evident that the direction of optimization with respect to $\pi_{\theta}(y_w | x)$ and $\pi_{\theta}(y_l | x)$, the longer response is likely to get bigger gradient. Fig. 5a and 5b show that under current settings the response length and logp are negative correlated, Fig. 5c shows that the data with larger length margin is more likely have a lower logp margin, this implies that the larger the length margin, the more the chosen response will play a role in gradient updates.

$$\frac{1}{\pi(y_w)} \nabla_{\theta} \pi(y_w) - \frac{1}{\pi(y_l)} \nabla_{\theta} \pi(y_l) = \frac{\nabla_{\theta} \pi(y_w)}{\prod_t^{|y_w|} \pi(y_w^{(t)} | y_w^{<t})} - \frac{\nabla_{\theta} \pi(y_l)}{\prod_t^{|y_l|} \pi(y_l^{(t)} | y_l^{<t})} \quad (8)$$

Liu et al. (2024) and Feng et al. (2024) suggest that a potential reason for verbosity in DPO is the im-

balanced gradient updates, which cause the model tend to generate longer chosen responses. Theoretically, even when a negative length bias exists in the dataset, where the chosen response is generally shorter than the rejected response, the model still tends to generate relatively longer responses. However, our results do not align with this expectation. We conduct additional exploration and report the findings in Appendix D.3.

Fig. 4c shows the result with complete mixed weight in different mixture coefficient settings. It is obvious that MWPO with mixed weight performs better compared to DPO, while also maintaining relatively shorter response length. The results show that using our mixed weight can achieve a better balance between performance and response length. We also leave some case examples in Appendix E.

6 Conclusion

In this work, we propose MWPO, an effective method that reweights preference pairs using multiple weights based on preference strength and length bias, respectively. Extensive experiments demonstrate that it can effectively control verbosity and improve overall performance. Furthermore, we analyze each weight to illustrate its necessity and

effectiveness. We also identify an inherent flaw in DPO that may lead to length exploitation, representing a potential avenue for future breakthroughs.

Limitations

While our method builds upon DPO and demonstrates promising overall performance with shorter generation length, it still has certain limitations:

- By leveraging the distribution feature of the dataset, we are able to control the generation length; however, this does not fundamentally address the length bias issue introduced by the accumulation calculation in DPO. Future work will be on how to eliminate length exploitation from the perspective of formulas.
- Due to resource constraints, our experiments are limited to four models. Further experiments are required to validate the scalability and generalization of the method across a broader range of models and datasets.
- Our method involves several hyperparameters that were manually set based on empirical experience. Among them, the control coefficient used to adjust constraint strength has a significant impact on the results. In the future, we plan to explore methods for reducing or automatically tuning these hyperparameters.

Acknowledgements

The authors sincerely thank the reviewers for their valuable comments, which improved the paper. The work is supported by the National Natural Science Foundation of China (62276057), and Sponsored by CAAI-MindSpore Open Fund, developed on OpenI Community.

References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- AI@Meta. 2024a. Llama 3 model card. https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.
- AI@Meta. 2024b. Llama 3.2 model card. https://github.com/meta-llama/llama-models/blob/main/models/llama3_2/MODEL_CARD.md.
- Anthropic. 2024. claude-3-5-sonnet. <https://www.anthropic.com/news/claude-3-5-sonnet>.
- Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi 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 (AISTATS)*, pages 4447–4455.
- Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, and et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.
- 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.
- 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:1803.05457v1*.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2024. Ultrafeedback: Boosting language models with high-quality feedback. In *Proceedings of the 2024 International Conference on Machine Learning (ICML)*.
- Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3029–3051.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, AhmadAl-Dahle, Aiesha Letman, and et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. 2024. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*.
- Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. Model alignment as prospect theoretic optimization. In *Forty-first International Conference on Machine Learning (ICML)*.

- 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 arXiv:2404.04626*.
- Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, and et al. 2024. A framework for few-shot language model evaluation. <https://github.com/EleutherAI/lm-evaluation-harness>.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, and et al. 2023a. Mistral 7b. *arXiv preprint arXiv:2310.06825*.
- Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, and et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.
- Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023b. LLM-blender: Ensembling large language models with pairwise ranking and generative fusion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 14165–14178.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles (SOSP)*.
- Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023a. AlpacaEval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval.
- Ziniu Li, Tian Xu, Yushun Zhang, Zhihang Lin, Yang Yu, Ruoyu Sun, and Zhi-Quan Luo. 2023b. ReMax: A simple, effective, and efficient reinforcement learning method for aligning large language models. In *Forty-first International Conference on Machine Learning (ICML)*.
- Wing Lian, Bley Goodson, Eugene Pentland, Austin Cook, Chanvichet Vong, and "Teknium". 2023. OpenOrca: An open dataset of gpt augmented flan reasoning traces. <https://huggingface.co/Open-Orca/OpenOrca>.
- Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 3214–3252.
- Wei Liu, Yang Bai, Chengcheng Han, Rongxiang Weng, Jun Xu, Xuezhi Cao, Jingang Wang, and Xunliang Cai. 2024. Length desensitization in directed preference optimization. *arXiv preprint arXiv:2409.06411*.
- 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. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1047–1067.
- Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems (NeurIPS)*, 35:27730–27744.
- Alizée Pace, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2024. West-of-n: Synthetic preference generation for improved reward modeling. In *ICLR 2024 Workshop on Navigating and Addressing Data Problems for Foundation Models*.
- Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. 2024. Disentangling length from quality in direct preference optimization. *arXiv preprint arXiv:2403.19159*.
- Silviu Pitis. 2023. Failure modes of learning reward models for llms and other sequence models. In *ICML 2023 Workshop The Many Facets of Preference-Based Learning*.
- Rafael Rafailov, Yaswanth Chittooru, Ryan Park, Harshit Sushil Sikchi, Joey Hejna, Brad Knox, Chelsea Finn, and Scott Niekum. 2024a. Scaling laws for reward model overoptimization in direct alignment algorithms. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pages 126207–126242.
- Rafael Rafailov, Joey Hejna, Ryan Park, and Chelsea Finn. 2024b. From r to q^* : Your language model is secretly a q-function. In *First Conference on Language Modeling*.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024c. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems (NeurIPS)*.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2023. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Yunhao Tang, Zhaohan Daniel Guo, Zeyu Zheng, Daniele Calandriello, Remi Munos, Mark Rowland, Pierre Harvey Richemond, Michal Valko, Bernardo Avila Pires, and Bilal Piot. 2023. Generalized preference optimization: A unified approach to offline alignment. In *Forty-first International Conference on Machine Learning (ICML)*.

Qwen Team. 2024. Qwen2.5: A party of foundation models. <https://qwenlm.github.io/blog/qwen2.5/>.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Al-lamabert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, and et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Cl  mentine Fourier, Nathan Habib, et al. 2023. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*.

Binghai Wang, Rui Zheng, Lu Chen, Yan Liu, Shihan Dou, Caishuang Huang, Wei Shen, Senjie Jin, Enyu Zhou, Chenyu Shi, et al. 2024a. Secrets of rlhf in large language models part ii: Reward modeling. *arXiv preprint arXiv:2401.06080*.

Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. 2024b. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. *arXiv preprint arXiv:2406.12845*.

Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. 2024. Contrastive preference optimization: Pushing the boundaries of llm performance in machine translation. In *Forty-first International Conference on Machine Learning (ICML)*.

Kevin Yang, Dan Klein, Asli Celikyilmaz, Nanyun Peng, and Yuandong Tian. 2024. Rlcd: Reinforcement learning from contrastive distillation for lm alignment. In *The Twelfth International Conference on Learning Representations (ICLR)*.

Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, and et al. 2024. Yi: Open foundation models by 01.ai. *arXiv preprint arXiv:2403.04652*.

Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. 2024. Rrhf: Rank responses to align language models with human feedback. *Advances in Neural Information Processing Systems (NeurIPS)*, 36.

Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J Liu. 2023. Slic-hf: Sequence likelihood calibration with human feedback. *arXiv preprint arXiv:2305.10425*.

Rui Zheng, Shihan Dou, Songyang Gao, Yuan Hua, Wei Shen, Binghai Wang, Yan Liu, Senjie Jin, Qin Liu, Yuhao Zhou, et al. 2023. Secrets of rlhf in large language models part i: Ppo. *arXiv preprint arXiv:2307.04964*.

Wenxuan Zhou, Ravi Agrawal, Shujian Zhang, Sathish Reddy Indurthi, Sanqiang Zhao, and et al. 2024. WPO: Enhancing RLHF with weighted preference optimization. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8328–8340.

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 arXiv:1909.08593*.

A Mathematical Derivations and Analysis

A.1 Formula Derivations

Due to the auto-regressive nature of LLM, given a prompt x , the probability of a response y under a policy π is:

$$\pi(y|x) = \prod_{t=1}^{|y|} \pi(y^{(t)} | x; y^{<t})$$

and the log probability is:

$$\log \pi(y|x) = \sum_{t=1}^{|y|} \log \pi(y^{(t)} | x; y^{<t})$$

where $|y|$ denotes the length of sequence y and $y^{<t}$ denotes all the tokens before the t -th token in y .

Finally, we get the token-level implicit reward formula without partition function as follows:

$$\begin{aligned} r(x, y) &= \beta \log \frac{\pi_{\theta}(y | x)}{\pi_{ref}(y | x)} \\ &= \beta \sum_{t=1}^{|y|} \log \frac{\pi_{\theta}(y^{(t)} | x; y^{<t})}{\pi_{ref}(y^{(t)} | x; y^{<t})} \end{aligned}$$

The theorem in Rafailov et al. (2024b) proves that given a reference policy π_{ref} and a parameter $\beta > 0$ all reward classes consistent with the

Bradley-Terry models can be represented with the a re-parameterization of the form as follow:

$$r(x, y) = \beta \log \pi(y | x) - \beta \log \pi_{ref}(y | x)$$

A.2 Theory Analysis

In the following, we will provide an analysis of the underlying principles of our approach from one plausible theoretical perspective.

We begin by adopting the definition of equivalent reward functions from the DPO paper: Two reward functions $r(x, y)$ and $r'(x, y)$ are equivalent iff $r(x, y) - r'(x, y) = f(x)$ for some function f .

Reward Equivalence. Given our linear parameterization $r(x, y) = \langle \phi(x, y), \theta^* \rangle$, we can demonstrate that the DPO reward:

$$r_{\text{dpo}}(x, y) = \beta \left[\langle \phi(x, y), (\theta - \theta_{\text{ref}}) \rangle - (\log Z_{\theta}(x) - \log Z_{\text{ref}}(x)) \right]$$

belongs to the same equivalence class as $r(x, y)$. The difference

$$r(x, y) - r_{\text{dpo}}(x, y) = \beta \log \frac{Z_{\theta}(x)}{Z_{\text{ref}}(x)}$$

depends solely on x , satisfying the equivalence criterion (Rafailov et al., 2024b).

Noise-Aware Preference Modeling. Assuming the true reward difference $r(x, y_1) - r(x, y_2)$ is corrupted by additive noise ϵ during annotation, preferences are generated by $(\langle \Delta \phi(x), \theta^* \rangle + \epsilon)$, where $\Delta \phi(x) = \phi(x, y_1) - \phi(x, y_2)$.

This leads to our Bradley-Terry negative log-likelihood objective:

$$\begin{aligned} \mathcal{L}_{\text{RM}}(\theta) = -\mathbb{E}_{x, \epsilon} \left[\log \sigma(\langle \Delta \phi(x), \theta \rangle) \right. \\ \cdot \mathbb{I}(\langle \Delta \phi(x), \theta^* \rangle + \epsilon > 0) \\ \left. + \log \sigma(-\langle \Delta \phi(x), \theta \rangle) \right. \\ \left. \cdot \mathbb{I}(\langle \Delta \phi(x), \theta^* \rangle + \epsilon \leq 0) \right] \end{aligned}$$

Optimality Analysis. Taking the gradient with respect to θ , the optimal solution satisfies:

$$\mathbb{E}_x [\sigma(\langle \Delta \phi(x), \theta \rangle)] = \mathbb{E}_{x, \epsilon} [\sigma(\langle \Delta \phi(x), \theta^* \rangle + \epsilon)]$$

Applying Jensen’s inequality to the concave $\sigma(\cdot)$ function:

$$\mathbb{E}_{x, \epsilon} [\sigma(\langle \Delta \phi(x), \theta^* \rangle + \epsilon)] \leq \mathbb{E}_x [\sigma(\langle \Delta \phi(x), \theta^* \rangle)]$$

This inequality implies θ must be scaled down relative to θ^* to maintain equality. Larger margin

examples where $\langle \Delta \phi(x), \theta^* \rangle$ is large mitigate the noise impact by reducing $\mathbb{E}_{\epsilon}[\sigma(\cdot + \epsilon)]$ sensitivity to ϵ . Data exhibiting a large reward margin during the training process typically indicates stronger preference strength, thus making it more likely to satisfy the conditions mentioned above.

B Training and Evaluation Details

For the Qwen2.5-1.5B-Base, we train the SFT models using the Ultrachat-200k dataset with the following hyperparameters: a learning rate of 1e-5, a maximum sequence length of 1024, a global batch size of 64, and a cosine learning rate scheduler with a 10% warmup ratio for one epoch. For the Llama-3.2-3B-Base model, we utilize an open-source SFT checkpoint trained on Ultrachat-200k⁴.

The main training hyperparameters and the hyperparameter search range for preference optimization are reported in Tables 3 and 4. All training experiments in this paper are conducted using the trl⁵ library and performed on $8 \times 910\text{b-64GB}$ Ascend NPUs.

For all datasets, we generate responses utilizing vllm (Kwon et al., 2023), an efficient framework for batched inference with LLMs. We sampling tokens with different temperature: 0.7 for AlpacaEval and 0.5 for Ultrafeedback testset.

For AlpacaEval 2, we use the official tool to evaluate on AlpacaEval 2⁶. The annotator is configured to use "weighted_alpaca_eval_gpt4_turbo", and "gpt4_turbo" for baseline generation.

For downstream benchmarks, we select different downstream tasks commonly used to conduct a more comprehensive assessment. We use Im-evaluation-harness (Gao et al., 2024) as the evaluation tool and evaluate in bfloat16 precision. The few-shot and batch settings as shown in Table 5.

C Correlation between GPT-4 and DeepSeek on LC Win-Rate of AlpacaEval 2

To ensure the effectiveness of the proxy annotator, we selected 9 different model architectures and sizes with open-source weights, along with AlpacaEval 2 evaluation results. The results are shown in Fig. 7, where the Spearman correlation between the DeepSeek-v3 results and GPT-4 re-

⁴<https://huggingface.co/tanliboy/llama-3.2-3b-sft>

⁵<https://github.com/huggingface/trl>

⁶https://github.com/tatsu-lab/alpaca_eval

	Mistral-7B-Base	Llama-3-8B-Instruct	Llama-3.2-3B-Base	Qwen2.5-1.5B-Base
Batch	2	2	8	16
Accumulations	8	8	2	1
Max Seq Len	1,024	1,024	1,024	1,024
Max Prompt Len	512	512	512	512
Lr	5e-7	8e-7	8e-7	1e-6
Warmup Ratio	0.1	0.1	0.05	0.05

Table 3: The main hyperparameters of MWPO.

Method	Hyperparameters Searching
DPO	$\beta \in \{0.1, 0.05, 0.01\}$
R-DPO	$\beta \in \{0.1, 0.05, 0.01\}$ $\alpha \in \{0.01, 0.05, 0.1\}$
SimPO	$\beta \in \{1, 2, 2.5, 5\}$ $\gamma \in \{0.1, 0.5, 1, 1.5\}$
MWPO	$\alpha \in \{0.4, 0.6, 0.8\}$ $\lambda \in \{0.01\}$

Table 4: The hyperparameter search range for Methods.

Type	Benchmark	Few-shot
MATH	GSM8K	8 CoT
World knowledge	TruthfulQA	0
Language Understanding	MMLU	5
Common sense reasoning	ARC-C	25
Disciplinary knowledge	GPQA	0

Table 5: The main settings of Downstream tasks.

sults is 1.00, demonstrating that the proxy annotator maintains excellent consistency.

D Exploration and Supplementary Experiments

D.1 Methods of Weight Mixture

The motivation for using *geometric* mean rather than *arithmetic* or *harmonic* mean is that geometric mixture is a common way of weight fusion between arithmetic and harmonic. The theoretical and experimental explanations as follows:

Balanced Contribution of Multiple Signals:

The geometric mean provides a balanced multiplicative fusion of the two signals. It elevates the combined weight only if both components are sufficiently large, reflecting a kind of joint contribution. Meanwhile, it is less extreme than the harmonic

mean in penalizing a single small weight, preserving a moderate influence from each signal.

Log-Space Smoothness: The geometric mean can be expressed as:

$$w_1^\alpha \cdot w_2^{1-\alpha} \implies \alpha \log(w_1) + (1 - \alpha) \log(w_2)$$

This log-space interpolation often yields smoother gradients during training compared to *arithmetic* and *harmonic*. It effectively avoids the tendency of arithmetic to be dominated by large values or harmonic to be overly penalized by very small values.

Experimental Results. We conduct experiments on Qwen2.5-1.5B-Base using different mixture methods and evaluate the outputs with AlpacaEval 2, judged by DeepSeek-V3. As shown in Table 6, the geometric mean outperforms both the arithmetic and harmonic means in improving the performance while effectively controlling the average generation length. This suggests that a multiplicative approach to combining the two weight signals is particularly effective in balancing reward disparities and generation-length variations under our experimental setting.

D.2 Experiments of Different λ_{lc}

Regarding the hyperparameter λ_{lc} , we did not perform searching for it, across all model settings, we fixed $\lambda_{lc} = 0.01$ simply, as it primarily serves to scale the length margin into a suitable range. We mainly consider that the length margins for the current dataset predominantly fall within the range of $-500 \sim 500$, as shown in Fig. 6a. The scaling coefficient $\lambda_{lc} = 0.01$ effectively normalizes these margins, ensuring that the differences between data are well-distinguished under the sigmoid function. This choice strikes a balance between sensitivity to length variations and numerical stability during optimization. Theoretically, excessive scaling weakens the effect of positive-margin data by pushing most values toward the sigmoid’s negative region, while moderate scaling yields a gentler transformation.

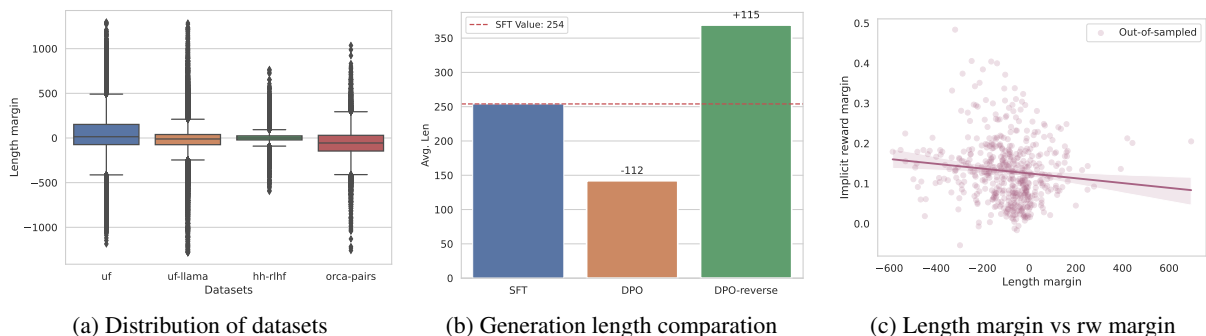


Figure 6: Function plots for weights under different constraint strengths. (a) Distribution of length margins for different preference datasets. (b) Generation length of DPO on dataset and its reverse. (c) The relation between the length margin and implicit reward margin.

	Geometric	Arithmetic	Harmonic
LC win-rate	12.87	10.24	10.17
Average Length	1656	1,705	1504

Table 6: AlpacaEval 2 results for different Weight mixture methods on Qwen2.5-1.5B-Base.

λ_{lc}	0.005	0.01	0.025	0.05
LC win-rate	11.21	12.87	9.75	1.72
Average Length	1839	1,656	1,326	227

Table 7: The results of different scale parameters.

Further, we add experimental results for the hyperparameter λ_{lc} . The results are shown in Table 7.

D.3 Experiments of Verbosity in DPO

As shown in Fig.6a, the preference datasets exhibit varying degrees of length bias, where uf indicates [HuggingFaceH4/ultrafeedback_binarized](#) (Cui et al., 2024), uf-llama indicates [princeton-nlp/llama3-ultrafeedback-armorm](#) (Meng et al., 2024), hh-rlhf indicates [Anthropic/hh-rlhf](#) (Bai et al., 2022). orca-pairs indicates [argilla/distilabel-intel-orca-dpo-pairs](#) (Lian et al., 2023).

To further verify the conclusion proposed by Liu et al. (2024) and Feng et al. (2024) as mentioned in Section 5.3, we train Qwen2.5-1.5B-Base using vanilla DPO on the orca-pairs dataset, where the chosen responses are generally shorter than the rejected responses. We then reverse the chosen and rejected columns in the dataset and train DPO again. We report the average generation length on AlpacaEval 2 in Fig.6b for both DPO and DPO-reverse. Surprisingly, DPO and DPO-reverse ex-

hibit almost identical length deviations. This suggests that if verbosity in DPO were due to a preference for longer chosen responses during gradient updates, the two datasets should not produce such similar length deviations. The length deviation of the dataset itself remains a significant factor, indicating that balancing this aspect during the construction of preference data in the future should be a priority.

Additionally, as shown in Fig.6c, since the length margin is negatively correlated with the reward margin under the current data distribution, the preference data with a small length margin is more likely to obtain a large reward-margin weight, making it easier for the model to learn the length preference.

E Case Studies

We present two examples from AlpacaEval 2 in Tables 8 and 9, where MWPO generates superior responses with fewer tokens compared to vanilla DPO.

In the first example, DPO’s response mentions irrelevant content such as "...are typically server-side languages like Python, Ruby, Java, or PHP that process server requests and handle data storage and retrieval." which are unrelated to the question. In the second example, DPO’s response contains excessive details about the Southern Ocean, which are not pertinent to the issue.

In both cases, MWPO’s responses accurately addressed the questions without unnecessary content, with language logic closely aligned with GPT-4.

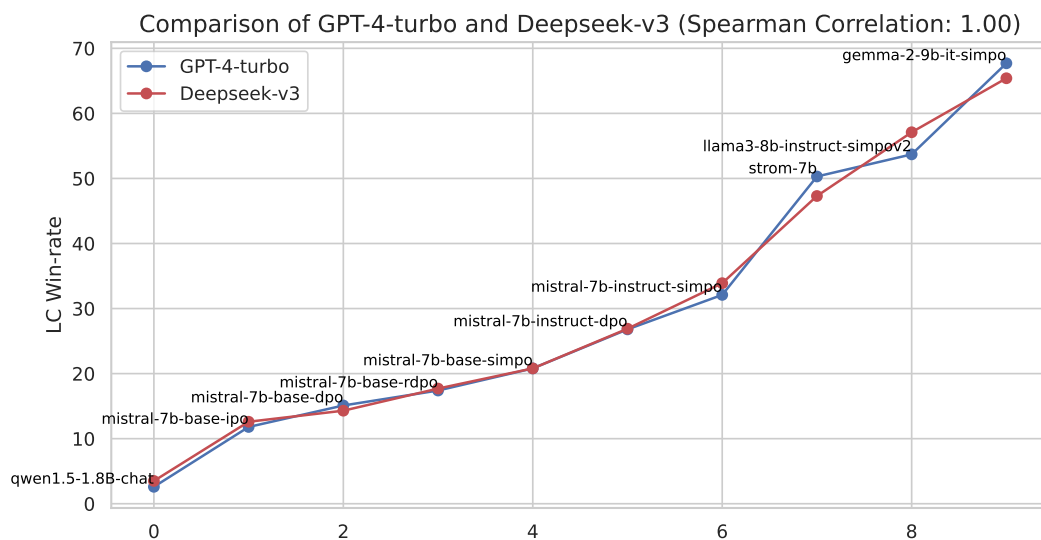


Figure 7: The correlation of LC win-rate between DeepSeek-v3 and GPT-4 on AlpacaEval 2.

Input	<p>Give a brief summary of the intention of the dialogue that just happened.</p> <p>Customer: Hi there, I'm looking for a new phone.</p> <p>AI: Hi! What type of phone are you looking for?</p> <p>Customer: I'm not sure. Maybe something with a good camera?</p> <p>AI: We have a few phones with great cameras. Would you like to see some options?</p> <p>Customer: Yeah, that would be great.</p>
DPO (480 tokens)	<p>The intention of the dialogue is for the customer, who is seeking a new phone, to receive recommendations from an AI for phones with good cameras based on their preferences. The AI aims to provide options for the customer to help them make an informed decision.</p>
MWPO (473 tokens)	<p>The dialogue between the customer and AI aims to guide the customer in finding a new phone with a focus on the camera feature. The AI provides suggestions and recommendations by presenting phone options with excellent camera capabilities.</p>
GPT-4	<p>The customer approached with the intention of finding a new phone, particularly one with a good camera. The AI, serving as a sales assistant or customer service representative, acknowledged the customer's need and offered to present several phone options that feature high-quality cameras. The customer expressed interest in seeing these options. The dialogue's purpose is to assist the customer in finding a suitable phone with a good camera.</p>

Table 8: Case examples of AlpacaEval 2, generated by GPT-4 and Mistral trained on DPO, MWPO.

Input	Give some examples of what people usually say in the given social situation. when someone arrives safely
DPO (819 tokens)	<ol style="list-style-type: none"> 1. "Welcome back! I'm so glad you made it safely." 2. "How was your journey? Everything went smoothly, I hope?" 3. "It's great to see you here unscathed!" 4. "Safe travels! I'm thrilled to hear you arrived in one piece." 5. "Your timely arrival is a testament to your safe and sound travel." 6. "Your safe arrival is a relief! How was your trip overall?" 7. "I'm delighted to see you safely here. Any interesting experiences during your travels?" 8. "Arrival without any mishaps! That's fantastic news!" 9. "Relief and cheer! We're overjoyed to have you here safely." 10. "It's a pleasure to see you in good health and spirits after a successful journey." 11. "Arrival without any hiccups! Congratulations!" 12. "Safe and sound is the best way to describe your arrival!" 13. "The journey was kind to you, and you've arrived safely. Bravo!" 14. "We're thrilled to see you in one piece after your travels." 15. "Congrats on arriving safely! Any stories to share?" 16. "Your safe arrival is a happy ending to your journey!" 17. "Wonderful to see you smiling, relaxed, and unscathed after your travels." 18. "The journey did not disappoint you, and you're safe. Amazing!" 19. "We're delighted to have you here safely. Your trip sounds fantastic." 20. "Arrival without a scratch, and we're overjoyed!"
MWPO (699 tokens)	<ol style="list-style-type: none"> 1. "Welcome back home! I'm so happy to hear you made it safely!" 2. "Great news! Congratulations on arriving safely. We've been holding our breaths until we heard." 3. "Ah, you must be relieved! Safe travels to you!" 4. "How wonderful! I'm so glad to hear you've arrived safely after your journey." 5. "Breathe a sigh of relief! You're here, and everything looks good." 6. "I'm so happy to see you, and you made it in one piece. That's fantastic!" 7. "You must have had some amazing experiences along the way, but first, let's celebrate your safe arrival!" 8. "Thank goodness! Your safety means everything to us." 9. "Your safe arrival brings us so much joy; please, share your travel tales!" 10. "You've made us proud with your safe arrival. Welcome back!" <p>Note: These examples cater to various styles and levels of formality.</p>
GPT-4	<p>When someone arrives safely at a destination, especially after a journey that might have involved some risk or concern, people often express relief and welcome. Here are some examples of what they might say:</p> <ol style="list-style-type: none"> 1. "I'm so glad you made it here safely!" 2. "Welcome! It's a relief to see you in one piece." 3. "Thank goodness you're here! We were starting to worry." 4. "Happy to see you've arrived without any trouble." 5. "It's great to see you! How was your trip?" 6. "I hope the journey wasn't too stressful for you." 7. "You had us worried for a bit, but we're so glad you're safe and sound." 8. "I've been checking the time, waiting for you to get here. Glad you're okay!" 9. "Seeing you walk through that door is the best part of my day!" 10. "I trust the travel was smooth? It's wonderful to have you here."

Table 9: Case examples of AlpacaEval 2, generated by GPT-4 and Mistral trained on DPO, MWPO.