SGDPO: Self-Guided Direct Preference Optimization for Language Model Alignment

Wenqiao Zhu †, 1, 2**, Ji Liu**†, 1, ***, Lulu Wang** 1**, Jun Wu**1**, Yulun Zhang**2
1 HiThink Research, 2 Shanghai Jiao Tong University

Abstract

Direct Preference Optimization (DPO) is broadly utilized for aligning Large Language Models (LLMs) with human values because of its flexibility. Despite its effectiveness, it has been observed that the capability of DPO to generate human-preferred response is limited and the results of DPO are far from resilient. To address these limitations, in this paper we propose a novel Self-Guided Direct Preference Optimization algorithm, i.e., SGDPO, which incorporates a *pilot* term to steer the gradient flow during the optimization process, allowing for fine-grained control over the updates of chosen and rejected rewards. We provide a detailed theoretical analysis of our proposed method and elucidate its operational mechanism. Furthermore, we conduct comprehensive experiments on various models and benchmarks. The extensive experimental results demonstrate the consistency between the empirical results and our theoretical analysis and confirm the effectiveness of our proposed approach (up to 9.19% higher score).

1 Introduction

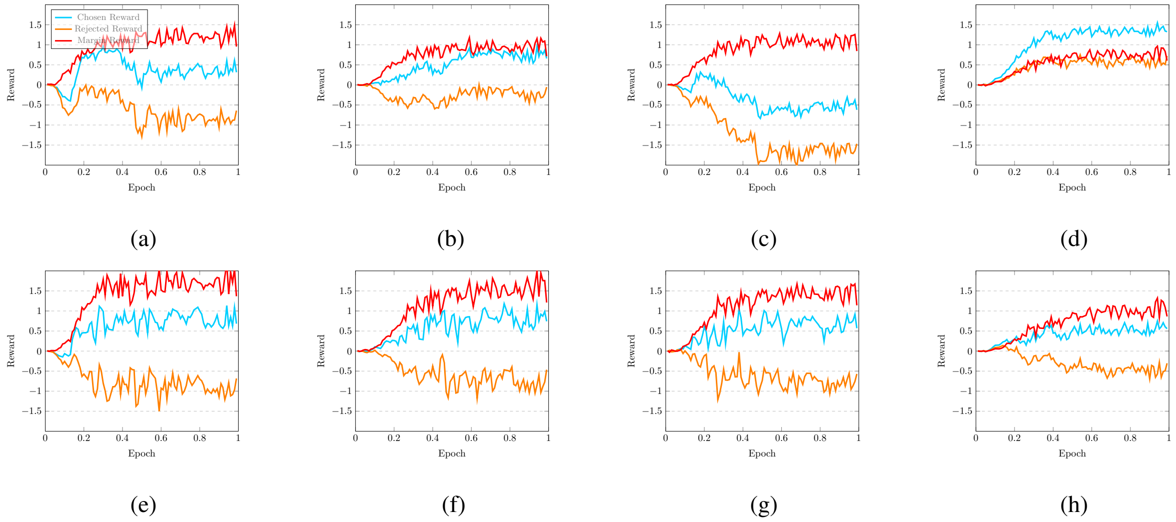
Large Language Models (LLMs) pretrained with next-token prediction have experienced rapid advancements (OpenAI, 2024; DeepSeek-AI et al., 2025; Anthropic, 2024; Google, 2024). This progress underscores the necessity to align LLM outputs with human values and preferences while safeguarding societal values from harm. Reinforcement Learning from Human Feedback (RLHF) has emerged as a critical method for achieving this alignment and has become an essential component within the LLM training pipeline (Stiennon et al., 2020; Bai et al., 2022; Bi et al., 2024).

Traditional RLHF typically involves three key steps: Supervised Fine-Tuning (SFT), reward learning, and Reinforcement Learning (RL) optimization. Because the RL optimization step relies heavily on the reward model, it is essential to train a high-quality reward model. However, this necessity adds complexity to the RLHF training process, making it intricate (Ilyas et al., 2020; Engstrom et al., 2020). To tackle this issue, Direct Preference Optimization (DPO) removes the need for reward training by reparameterizing the reward model (Rafailov et al., 2023). Specifically, it maps reward functions to optimal policies by employing the Bradley-Terry Model (Bradley and Terry, 1952a), thereby transforming preference feedback from online reward models into offline implicit modeling. As a result, DPO simplifies the post-training process.

While it has been widely adopted for its flexibility with similar performance levels compared to classic RLHF methods, e.g., PPO (Dubois et al., 2023), ChatGLM-RLHF (Hou et al., 2024), the limitations of DPO are observed in a bunch of investigation, which lead to suboptimal alignment performance in LLM training. These limitations include high computational costs (Ethayarajh et al., 2024; Hong et al., 2024; Meng et al., 2024), verbosity (Park et al., 2024; Liu et al., 2024f; Lu et al., 2024), and overfitting (Azar et al., 2023; Jung et al., 2024a; Gheshlaghi Azar et al., 2024). In addition, DPO may still incur inferior capability of LLMs in producing responses that resonate with human preferences (Feng et al., 2024). While LLMs trained with DPO tend to avoid generating responses humans dislike, they struggle to generate responses that humans prefer. Furthermore, the efficacy of DPO is inconsistent while being sensitive to the effectiveness of Supervised Fine-Tune (SFT) (Feng et al., 2024; Xu et al., 2024). For instance, LLMs with improper and ineffective settings may lead to poor DPO performance. As illustrated in Figures 1 (a), (b), (c), and (d), the training reward curves of DPO on various base models using the same

[†] Equal Contribution
[*] Corresponding author: jiliuwork@gmail.com

Findings of the Association for Computational Linguistics: ACL 2025, pages 12366–12383
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 1: Reward curves on various base models: (a) DPO reward curves on Llama-3.1 instruct 8B; (b) DPO reward curves on Llama-3.1 base 8B; (c) DPO reward curves on Qwen-2 instruct 7B; (d) DPO reward curves on Qwen-2 base 7B. (e) Our SGDPO reward curves on Llama-3.1 instruct 8B; (f) Our SGDPO reward curves on Llama-3.1 base 8B; (g) Our SGDPO reward curves on Qwen-2 instruct 7B; (h) Our SGDPO reward curves on Qwen-2 base 7B.

preference dataset exhibit extremely high diversity.

Recently, some theoretical works (Pal et al., 2024; Feng et al., 2024) reveal the reasons behind the limitations of DPO. First, the standard DPO loss can lead to a reduction in the likelihood of preferred examples generated by LLMs (Pal et al., 2024), especially when the Hamming distance between preferred and dispreferred responses is low. Second, the limitations of DPO may be attributed to undesired distinct update patterns in gradient flow between chosen and rejected rewards (Feng et al., 2024). When the optimization process enters an undesired region, the gradient flow of DPO tends to generate an unbalanced update to different variables or incurs difficulties in escaping saddle points, leading to inferior optimization performance.

In this paper, we propose a novel Self-Guided Direct Preference Optimization algorithm, i.e., SGDPO, to address the aforementioned limitations. We introduce a *pilot* term into the objective function of SGDPO. This *pilot* term can be adjusted to steer gradient updates towards different regions, resulting in diverse gradient update patterns and consequently leading to distinct optimization processes. In this case, SGDPO can enhance the alignment capability of LLMs to generate responses preferred by humans, while contributing to the stabilization and resilience of the LLM training process, as well. In addition, we carry out a detailed theoretical analysis to illustrate the robustness and resilience of SGDPO. Furthermore, we conduct extensive experiments across various models and benchmarks to demonstrate the superb performance of SGDPO. The major contributions are summarized as follows:

- We propose a novel preference alignment algorithm, i.e., Self-Guided Direct Preference Optimization (SGDPO), designed to stabilize the LLM training process and enhance the capability of LLMs so as to generate responses preferred by humans. By incorporating a *pilot* term into the objective function, SGDPO guides the gradient flow to balanced updates, thereby improving the updates of chosen and rejected rewards.

- We provide a thorough theoretical analysis of SGDPO, elucidating its underlying mechanisms for its robustness and resilience. This analysis offers a scheme for controlling the updates of chosen and rejected rewards as well.

- We conduct extensive experiments across 4 models and 8 benchmarks. Experimental results demonstrate the alignment between our theoretical analysis and empirical observation, which validates the effectiveness of SGDPO. Specifically, our method has achieved a significant improvement over the DPO method, with the relative increase reaching up to a maximum of 9.19%.

2 Related Work

RLHF has been proven effective in aligning LLMs with human values and has seen widespread adoption across various applications, e.g., summarization (Stiennon et al., 2020), safety alignment (Bai et al., 2022), instruction following (Ouyang et al., 2022), and translation (Xu et al., 2024). Nevertheless, RLHF requires a complex training pipeline, which has spurred the proposal of DPO (Rafailov et al., 2023) to simplify the LLM training pipeline.

Since the introduction of DPO, a variety of extensions have been proposed to either address its limitations or provide theoretical interpretations. These include new preference optimization techniques (Xiao et al., 2024; Zeng et al., 2024; Razin et al., 2025) and analytical studies (Pal et al., 2024; Feng et al., 2024). For instance, SimPO (Lu et al., 2024) reduces computational overhead by adopting a reference-free training strategy, while SimPER (Xiao et al., 2025) introduces an inverse perplexity objective to lower the complexity and fine-tuning time of large language models (LLMs). Although SimPER results in a smaller decrease in chosen likelihoods compared to SimPO, it still exhibits a declining trend in chosen likelihoods, indicating limited flexibility. In contrast, our method introduces a mechanism that allows for adjustable control over both chosen and rejected likelihoods, thereby offering greater adaptability.

Similarly, SamPO and LD-DPO (Liu et al., 2024f) aim to reduce the verbosity often introduced by alignment algorithms due to prior biases in preference data, ultimately improving alignment performance. TDPO (Zeng et al., 2024) enhances alignment and diversity through a token-level optimization approach. IPO (Gheshlaghi Azar et al., 2024) mitigates overfitting by introducing a regularization term that pulls the solution toward a reference policy. Cal-DPO (Xiao et al., 2024), on the other hand, improves performance by incorporating absolute reward values instead of relying solely on relative ones—similar in spirit to IPO's regularization goal. However, in many practical scenarios, exact absolute reward values may not be available, requiring approximations that can lead to suboptimal outcomes. Our method avoids this issue entirely, as it does not depend on absolute reward values and instead provides additional flexibility through tunable parameters for both chosen and rejected likelihoods, as well as their ratios.

Unintentional Unalignment (Razin et al., 2025) investigates how similar embeddings from preference data can lead to unintended misalignment. The authors introduce a metric called Centered Hidden Embedding Similarity (CHES) to improve training sample selection. While this approach is promising for dataset curation, our method operates at the optimization level rather than the data level, making it more robust and independent of dataset modifications. Additionally, NCA (Chen et al., 2024) leverages Noise Contrastive Estimation (NCE) to achieve robust alignment, and BCO (Jung et al., 2024b) proposes training a binary classifier where the logit serves as a reward signal, also yielding robust results.

Despite these advances, none of the existing methods effectively tackle both the issue of reduced updates on preferred examples and the challenge of unbalanced updates with difficulties in escaping saddle points simultaneously. Our approach addresses both concerns, offering a more comprehensive and flexible solution to preference-based alignment.

3 Method

3.1 Preliminary of DPO

DPO is a widely adopted technique for optimizing the preferences of LLMs. This method stands out because of the innovative utilization of an analytical mapping that translates reward functions into optimal policies, streamlining the alignment process without necessitating a direct reward model. The cornerstone of DPO lies in its specific transformation, which can be mathematically formulated by the following equation:

$$r(x,y) = \beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)} + \beta \log Z(x), \qquad (1)$$

where $r(x,y)$ is the reward function, β serves as a scaling factor, $\pi_\theta(y|x)$ represents the policy inferred from the reward model, and $\pi_{\text{ref}}(y|x)$ indicates the reference policy. Here, $Z(x)$ functions as a normalization constant ensuring the probabilities are properly scaled.

By leveraging the Bradley-Terry preference model (Bradley and Terry, 1952b), DPO expresses the probability that chosen outcome y_w is preferred over rejected outcome y_l, given an input prompt instruction x, as formulated as follows:

$$p(y_w > y_l|x) = \frac{\exp\left(r(x,y_w)\right)}{\exp\left(r(x,y_w)\right) + \exp\left(r(x,y_l)\right)}. \qquad (2)$$

The Formula 2 quantifies the relative preference between two responses by comparing their associated reward values. Within this probabilistic framework, the loss function of DPO, denoted by $\mathcal{L}_{DPO}$, is formulated as Formula 3:

$$\mathcal{L}_{DPO} = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[l_{\text{DPO}}(\pi_\theta, \pi_{\text{ref}})\right], \quad (3)$$

where $l_{\text{DPO}}(\pi_\theta, \pi_{\text{ref}})$ is defined by:

$$l_{\text{DPO}}(\pi_\theta, \pi_{\text{ref}}) = \log\sigma(\Delta) \quad (4)$$

Here

$$\Delta = \beta\left[\log\frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \log\frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\right] \quad (5)$$

σ represents the sigmoid function, and β serves as a scaling factor as that in Formula 1. Formulas 3 and 4 thereby encapsulate the principles of the Bradley-Terry model, integrating preference data into the learning process. In this way, DPO ensures that the responses of LLMs align with observed human preferences.

3.2 Optimization Process of DPO

Given the chosen reward $\mathcal{X}_1 = \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)}$ and the rejected reward $\mathcal{X}_2 = \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}$, the partial derivatives of l_{DPO} with respect to $\mathcal{X}_1$ and $\mathcal{X}_2$ are calculated in Formulas 6 and 7 (Feng et al., 2024):

$$\frac{\partial l_{\text{DPO}}}{\partial \mathcal{X}_1} = \frac{\beta\mathcal{X}_2^\beta}{\mathcal{X}_1(\mathcal{X}_1^\beta + \mathcal{X}_2^\beta)}, \quad (6)$$

$$\frac{\partial l_{\text{DPO}}}{\partial \mathcal{X}_2} = -\frac{\beta\mathcal{X}_2^{\beta-1}}{\mathcal{X}_1^\beta + \mathcal{X}_2^\beta}. \quad (7)$$

Furthermore, the ratio of the increase in the probability of a human-preferred response to the decrease in the probability of a human-dispreferred response is given by:

$$\left|\frac{\partial l_{\text{DPO}}/\partial \mathcal{X}_1}{\partial l_{\text{DPO}}/\partial \mathcal{X}_2}\right| = \frac{\mathcal{X}_2}{\mathcal{X}_1} \quad (8)$$

The DPO gradient flow for the chosen and rejected rewards are shown in Figure 2. Based on the theoretical framework outlined above and this figure, we can make the following observations:

- When $\mathcal{X}_2$ is small, as illustrated in the lower part of Figure 2, the DPO gradient flow tends to decrease $\mathcal{X}_2$ rapidly while making only minor adjustments to $\mathcal{X}_1$. This behavior limits the ability of LLMs to effectively generate highly preferred responses.

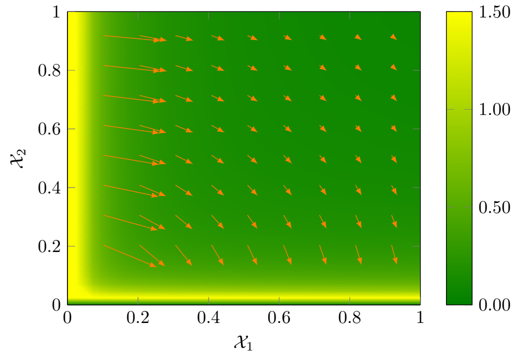


Figure 2: Gradient flow of DPO ($\beta = 0.1$) with large values truncated at 1.5.

- As DPO optimization progresses, the chosen reward $\mathcal{X}_1$ increases while the rejected reward $\mathcal{X}_2$ decreases. Consequently, $\frac{\mathcal{X}_2}{\mathcal{X}_1} < 1$. According to Equation 8, this results in the gradient for the rejected reward being updated more quickly than that for the chosen reward.

3.3 SGDPO

The theoretical framework discussed above suggests several directions for improving DPO:

- **G1:** Prevent the rejected reward $\mathcal{X}_2$ from rapidly dropping to a very small value, which otherwise halts meaningful updates to the chosen reward $\mathcal{X}_1$ or enhance the gradient update of chosen rewards $\mathcal{X}_2$.

- **G2:** Increase the ratio in Equation 8 to allow for more substantial updates to the chosen reward, thereby enhancing the capability of LLMs to generate preferred responses.

These adjustments aim to refine the optimization process of RLHF and enhance the performance of LLM in aligning with human preferences.

To achieve the aforementioned goals, we propose incorporating an adjusted preference optimization objective in the loss function of SGDPO as defined in Formula 9:

$$\mathcal{L}_{pilot} := -\frac{1}{2}\mathbb{E}_{(x,y_w,y_l\sim\mathcal{D})}\left[l_{pilot}(\pi_\theta, \pi_{\text{pilot}})\right], \quad (9)$$

where $l_{pilot}(\pi_\theta, \pi_{\text{pilot}})$ is defined in Formula 10.

$$\begin{aligned} & l_{pilot}(\pi_\theta, \pi_{\text{pilot}}) := \\ & \log \sigma \left(\beta \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \log \frac{\pi_{\text{pilot}}(\hat{y}_l|x)}{\pi_{\text{ref}}(\hat{y}_l|x)} \right) \\ & + \log \sigma \left(\beta \log \frac{\pi_{\text{pilot}}(\hat{y}_w|x)}{\pi_{\text{ref}}(\hat{y}_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)} \right), \end{aligned} \tag{10}$$

where $\hat{y}_w$ and $\hat{y}_l$ denote the sub-sequences of y_w and y_l, respectively. See details for the construction of $\hat{y}_w$ and $\hat{y}_l$ in Section 3.4. In order to simplify the calculations, we introduce $\mathcal{Y}_1 = \frac{\pi_{\text{pilot}}(\hat{y}_w|x)}{\pi_{\text{ref}}(\hat{y}_w|x)}$ and $\mathcal{Y}_2 = \frac{\pi_{\text{pilot}}(\hat{y}_l|x)}{\pi_{\text{ref}}(\hat{y}_l|x)}$. Let us denote the length of the token sequence of y by T, with y_t representing the token at the t-th index and $y_{<t}$ denoting all tokens preceding the t-th index. Given that $\hat{y}$ is a subsequence of y, and considering $\pi_\star(y|x)$ can be represented as $\prod_{t=1}^{T} \pi_\star(y_t|y_{<t}, x)$ on a token level basis, where $\star$ belongs to the set {pilot, ref, θ}, we can express $\mathcal{X}_2 = p_2\mathcal{Y}_2$ and $\mathcal{X}_1 = p_1\mathcal{Y}_1$, where $\mathcal{X}_1$ and $\mathcal{X}_2$ are defined in Section 3.2. p_1 and p_2 represent the product of the token probability ratios for the remaining tokens in sequences $\mathcal{X}_1$ and $\mathcal{X}_2$, excluding the sub-sequences $\mathcal{Y}_1$ and $\mathcal{Y}_2$.

The π_{pilot} in Formula 10 is the guiding policy model to steer the reward updates, we then demonstrate the advantages of the adjusted preference optimization objective to be leveraged to enhance preference optimization:

Theorem 1. *The partial derivatives of l_{pilot} with respect to $\mathcal{X}_1$ and $\mathcal{X}_2$ are given by:*

$$\frac{\partial l_{pilot}}{\partial \mathcal{X}_1} = \frac{\beta \mathcal{Y}_2^\beta}{\mathcal{X}_1(\mathcal{X}_1^\beta + \mathcal{Y}_2^\beta)} \tag{11}$$

$$\frac{\partial l_{pilot}}{\partial \mathcal{X}_2} = -\frac{\beta \mathcal{X}_2^{\beta-1}}{\mathcal{Y}_1^\beta + \mathcal{X}_2^\beta} \tag{12}$$

Proof. We defer the detailed proof to Appendix A.1. □

From Theorem 1, we can observe that the gradients of $\mathcal{X}_1$ and $\mathcal{X}_2$ depend on $\mathcal{Y}_1$ and $\mathcal{Y}_2$, respectively. Consequently, by manipulating $\mathcal{Y}_1$ and $\mathcal{Y}_2$, we can control the gradient flow within the alignment method, thereby influencing the updates to the chosen and rejected rewards. We present the visualized representation of these functions in Figure 10 in the Appendix.

Theorem 2. *The partial derivative $|\frac{\partial l_{pilot}}{\partial \mathcal{X}_1}|$ increases as $\mathcal{Y}_2$ increases, while the partial derivative $|\frac{\partial l_{pilot}}{\partial \mathcal{X}_2}|$ decreases as $\mathcal{Y}_1$ increases.*

Proof. Please see detailed proof in Appendix A.1. □

As preference alignment algorithms enhance the generation probability of preferred text while diminishing that of non-preferred text during fine-tuning, we have $p_2 < 1$. Consequently, $\mathcal{Y}_2 > \mathcal{X}_2$. Comparing Formula 11 with Formula 6, we can see that the difference lies in just one variable. For instance, $\mathcal{X}_2$ in Formula 6 is replaced with $\mathcal{Y}_2$ to derive Formula 11. Based on Theorem 2, we then have:

$$\left| \frac{\partial l_{\text{pilot}}}{\partial \mathcal{X}_1} \right| > \left| \frac{\partial l_{\text{DPO}}}{\partial \mathcal{X}_1} \right| \tag{13}$$

Formula 13 reveals that SGDPO can enlarge the gradient of chosen rewards, which enhances the updating of chosen rewards. Consequently, SGDPO boosts the generation of preferred responses.

Theorem 3. *Let $\pi_{pilot} = \pi_\theta$ and $z = \frac{\mathcal{Y}_1}{\mathcal{Y}_2}$, for each pairwise preference instance $(x, y_w, y_l) \in \mathcal{D}$, the ratio between the increase in the probability of a human-preferred response and the decrease in the probability of a human-dispreferred response is given by:*

$$\left| \frac{\partial l_{pilot}}{\partial \mathcal{X}_1} / \frac{\partial l_{pilot}}{\partial \mathcal{X}_2} \right| = \frac{\mathcal{X}_2}{\mathcal{X}_1} \cdot f(z), \tag{14}$$

where

$$f(z) = \frac{1}{p_2^\beta} \frac{z^\beta + p_2^\beta}{p_1^\beta z^\beta + 1}. \tag{15}$$

$f(z)$ is a monotonic function of z. When $p_1p_2 < 1$, the function $f(z)$ is a monotonically increasing function of z. Conversely, when $p_1p_2 > 1$, the function $f(z)$ is a decreasing function of z. Furthermore, $f(z) > 1$ when $p_1p_2 < 1$.

Proof. Please see detailed proof in Appendix A.1. □

When the rejected reward decreases rapidly, it leads to $p_1p_2 < 1$. Consequently, this results in $f(z) > 1$, which boosts the ratio value given by Equation 14. As $z = \frac{\mathcal{Y}_1}{\mathcal{Y}_2}$ increases, $f(z)$ also increases throughout the training process. This behavior aligns with our goal G2, thereby enhancing the capability of LLM to generate preferred text. We present the visual representation of $f(z)$ in Figure 9 in the Appendix. A comparison between Figure 1 (c) and (g) illustrates an example of how Theorem 3 works, wherein both DPO and SGDPO are trained on the same base model using the same preference dataset.

3.4 Sub-sequence Construction

Based on Theorems 1, 2, and 3, we derive sub-sequences $\hat{y}_w$ and $\hat{y}_l$ from the sequences y_w and y_l, respectively. $\hat{y}_w$ and $\hat{y}_l$ serve as indicators to guide the refinement of updates for chosen and rejected reward adjustments. Let l_1 and l_2 denote the lengths of the sequences y_w and y_l. We define l_c as the minimum length between l_1 and l_2. From the pairs (y_w, y_l), we randomly select preference data pairs $(\hat{y}_w, \hat{y}_l)$ with lengths $(r_1 \cdot l_c, r_2 \cdot l_c)$, where r_1 and r_2 are hyper-parameters.

While generating $\hat{y}_w$ and $\hat{y}_l$ for both the *pilot* model and the reference model, we can exploit the same random index or different random indices. Utilizing the same random index ensures that the sub-sequences are constructed from an identical set of tokens. Conversely, employing different random indices results in sub-sequences derived from distinct sets of tokens. We refer to the setting with the same index as $Pilot_s$ and that with different indices as $Pilot_d$. As different random indices may introduce an element of randomness into the learning space so as to allow SGDPO to explore more thoroughly and avoid overfitting with superb performance, we exploit $Pilot_d$ in SGDPO.

Adjusting r_1 and r_2 is critical to the gradient changes associated with the chosen and rejected rewards during the preference optimization process, as indicated by Theorem 2. Smaller values of r_2 lead to shorter *pilot* sequences, which in turn increases $\mathcal{Y}_2$. As the model converges, the likelihood of encountering tokens from $\hat{y}_l$ decreases. Hence, decreasing r_2 typically leads to larger magnitudes of the partial derivatives $\left|\frac{\partial l_{\text{pilot}}}{\partial \mathcal{X}_1}\right|$. Then, the chosen rewards are updated rapidly and the capability of generating human-preferred responses is improved. However, to conserve the semantic meanings of responses, we empirically set $r_1 \geq 0.6$ and $r_2 \geq 0.6$. Moreover, some randomness may exist in the sub-sequence construction and training process, thus, we fine-tune the values of r_1 and r_2 to achieve superb performance compared with that of $r_1 = 0.6$ and $r_2 = 0.6$ (see details in Section 4.3.2).

4 Experimental Evaluation

In this section, we compare SGDPO with 6 state-of-the-art performance optimization algorithms, exploiting 4 model configurations and 8 tasks. We first present the experimental setup. Then, we illustrate the experimental results. Finally, we show the ablation study.

4.1 Experimental Setup

We compare SGDPO against 6 baselines, including DPO (Rafailov et al., 2023), SamPO (Lu et al., 2024), IPO (Gheshlaghi Azar et al., 2024), Token-Level DPO (Zeng et al., 2024), NCA (Chen et al., 2024), and BCO (Jung et al., 2024b). These competitive baselines cover a broad range of methods, addressing issues such as eliminating verbosity, avoiding overfitting, ensuring robust alignment, and more. In addition, we consider Llama-3.1 8B (AI@Meta, 2024) and Qwen-2 7B (Yang et al., 2024) across two configurations: Instruct and Base, which corresponds to 4 model configurations. For the Instruct configuration, we use the instructed model as the Supervised Fine-Tuned (SFT) model, which has already undergone a Supervised Fine-Tuning phase. In contrast, for the Base configuration, we fine-tune the base model with the UltraChat-200k dataset (Ding et al., 2023) to create the SFT model, which enhances the base LLM capacity to follow instructions. We leverage the publicly available UltraFeedback dataset (Cui et al., 2023) as human preference data. Each entry in the UltraFeedback dataset follows the format (x, y_w, y_l), designed to reflect human values such as helpfulness and honesty.

We exploit two open-ended generation benchmarks, i.e., MT-Bench (Zheng et al., 2023) and AlpacaEval-2 (Li et al., 2023; Dubois et al., 2024) (see details in Appendix). For the conditional benchmarks, we evaluate our models on the following 6 tasks: MMLU in a 5-shot setting (Hendrycks et al., 2021), GSM8K in an 8-shot setting (Cobbe et al., 2021), PiQA in a 3-shot setting (Bisk et al., 2020), TruthfulQA in a 3-shot setting (Lin et al., 2022), IFEVAL in a 3-shot setting (Zhou et al., 2023), and ARC in a 3-shot setting (Clark et al., 2018). Please see details of the experimental setup in Appendix A.2.

4.2 Evaluation of SGDPO

In this section, we present the experimental comparison of SGDPO with 7 state-of-the-art optimization algorithms. We first present the experimental results on two open-ended benchmarks, i.e., MT-Bench and AlpacaEval-2. Then, we show the results on conditioned benchmarks. Finally, we present the training rewards of SGDPO compared with DPO.

As shown in Table 1, SGDPO achieves the highest average score compared to other approaches

Methods	Llama-3.1 instruct 8B				Llama-3.1 base 8B			
	Score_1	Score_2	$\text{Score}_{\text{avg}}$	$\text{Token}_{\text{len}}$	Score_1	Score_2	$\text{Score}_{\text{avg}}$	$\text{Token}_{\text{len}}$
SFT	8.40	7.54	7.97	287	7.47	6.60	7.03	247
DPO (Rafailov et al., 2023)	8.27	7.36	7.82	329	7.29	6.41	6.85	263
NCA (Chen et al., 2024)	8.13	7.21	7.67	308	7.47	6.75	7.11	256
BCO (Jung et al., 2024b)	8.22	7.25	7.74	305	7.44	6.43	6.94	278
IPO (Gheshlaghi Azar et al., 2024)	**8.57**	7.51	8.04	383	7.51	**7.00**	7.26	255
SamPO (Lu et al., 2024)	8.34	7.66	8.00	289	7.70	6.52	7.11	262
TDPO (Zeng et al., 2024)	8.39	7.37	7.88	296	7.30	6.38	6.84	268
SGDPO	8.38	**7.90**	**8.14**	312	**7.98**	6.90	**7.44**	264
	Qwen-2 instruct 7B				Qwen-2 base 7B			
	Score_1	Score_2	$\text{Score}_{\text{avg}}$	$\text{Token}_{\text{len}}$	Score_1	Score_2	$\text{Score}_{\text{avg}}$	$\text{Token}_{\text{len}}$
SFT	8.14	7.64	7.78	311	7.94	6.80	7.37	269
DPO (Rafailov et al., 2023)	8.44	7.99	8.21	307	7.87	6.99	7.43	293
NCA (Chen et al., 2024)	8.41	**8.12**	8.27	303	7.83	7.34	7.58	291
BCO (Jung et al., 2024b)	8.49	7.97	8.23	309	7.87	6.62	7.25	326
IPO (Gheshlaghi Azar et al., 2024)	8.31	8.04	8.17	312	7.65	**7.42**	7.54	351
SamPO (Lu et al., 2024)	8.56	7.86	8.21	307	8.09	7.09	7.59	320
TDPO (Zeng et al., 2024)	8.32	7.94	8.13	313	7.91	6.88	7.39	327
SGDPO	**8.68**	8.04	**8.36**	318	**8.26**	7.09	**7.67**	329

Table 1: MT-Bench Results across different model configurations. Here, Score_1 refers to the score from the first turn, Score_2 to the score from the second turn, and $\text{Score}_{\text{avg}}$ represents the average score. $\text{Token}_{\text{len}}$ indicates the average length of output tokens for each method. we set $r_1 = r_2$ for SGDPO in this experiment.

Method	GSM8K	MMLU	PiQA	TruthfuQA	IFEval	ARC	Avg.
SFT	0.5625	0.7060	0.8096	0.5734	**0.4251**	0.8582	0.6558
DPO (Rafailov et al., 2023)	0.5989	0.7065	**0.8112**	0.5774	0.4140	0.8628	0.6618
NCA (Chen et al., 2024)	0.5921	0.7057	0.8079	0.5782	0.4140	0.8607	0.6598
BCO (Jung et al., 2024b)	0.5898	0.7065	0.8074	0.5776	**0.4251**	0.8620	0.6614
IPO (Gheshlaghi Azar et al., 2024)	**0.6406**	0.7039	0.7894	**0.5876**	0.3974	0.8535	0.6620
SamPO (Lu et al., 2024)	0.6133	0.7067	0.8074	0.5844	0.3993	0.8632	0.6623
TDPO (Zeng et al., 2024)	0.5951	0.7055	0.8089	0.5763	0.3967	0.8589	0.6569
SGDPO	0.6111	**0.7069**	0.8107	0.5806	0.4196	**0.8641**	**0.6655**

Table 2: Evaluation results on conditional benchmarks for various approaches, using Qwen-2 instruct 7B as the base model.

with the MT-Bench benchmark across various base models. Specifically, SGDPO significantly outperforms DPO (from 1.83%, to 8.61%), which highlights the broad applicability of our proposed method across different base models and confirms its effectiveness through high average scores. Moreover, the table reveals that DPO does not invariably enhance the MT-Bench score, which is in line with previous findings (Liu et al., 2024f). This result can be attributed to the limitations of DPO as discussed in Section 3.2. In addition, compared to the SFT baseline, most alignment methods tend to produce longer response lengths. Notably, the response length of SGDPO is similar to that of DPO with negligible length bias brought by the *pilot* term, e.g., SGDPO has a shorter response length on Llama-3.1 instruct 8B, while it has a longer response length on Qwen-2 instruct 7B compared to DPO. Furthermore, the experimental results confirm the capability of SGDPO to escape saddle points. IPO and SamPO have similar performance while their response lengths differ significantly. Meanwhile, the average scores of IPO and SamPO are lower than that of SGDPO, which indicates that SamPO and IPO may become trapped in different local optima. In contrast, SGDPO utilizes a self-guide scheme to avoid getting trapped in a suboptimal policy. In addition, on AlpacaEval-2 benchmark, our experimental results show that SGDPO outperforms DPO by 2.51% on the LC win rate metrics when evaluating the Llama-3.1 instruct 8B model (see details in Appendix).

As shown in Table 2, SGDPO achieves the highest average score (up to 0.0097) compared with

7 competitive baselines based on the conditional benchmarks. In addition, the experimental results demonstrate that all alignment algorithms improve the average score when compared to the SFT baseline. This implies that these alignment algorithms can enhance the capabilities of LLMs to a certain extent. IPO and SamPO achieve higher scores on the GSM8K benchmark, which may suggest that avoiding overfitting and eliminating length bias could improve the reasoning abilities of LLMs. From Tables 1 and 2, we can also observe that the performance of different algorithms varies between open-ended and conditional benchmarks. Hence, different alignment algorithms correspond to diverse capability aspects of LLMs.

In order to show the robust performance of SGDPO, we present the training awards with diverse model configurations in Figure 1. In this experimentation, the training reward curve for SGDPO was generated using hyper-parameters r_1 and r_2, both set to 0.6. The figure demonstrates that SGDPO is much more stable than DPO across all the base model configurations. We empirically observe that the patterns of the training rewards for DPO vary significantly across different base models. For instance, the chosen reward of DPO on Llama-3.1 instruct 8B first increases, then drops to a low value, while the chosen reward of DPO on Qwen-2 instruct 7B shows the well-observed decreasing-likelihood phenomenon. In contrast, SGDPO exhibits consistent reward patterns. These findings reveal that SGDPO offers greater resilience compared to the DPO method.

4.3 Ablation Study

In this section, we first present experimental results for selecting between *Pilot*$_s$ and *Pilot*$_d$. Next, we analyze the impact of the hyper-parameters r_1 and r_2 on *Pilot* and overall model performance. We further investigate the behavior of SGDPO under settings where $r_1 \neq r_2$. Finally, we compare SGDPO with ORPO, a recent method for preference optimization.

4.3.1 Sub-sequence Construction

As shown in Figure 4, we carry out an experiment for the comparison between *Pilot*$_s$ and *Pilot*$_d$ with Llama-3.1 instruct 8B and MT-Bench. The experimental results demonstrate that both *Pilot*$_s$ and *Pilot*$_d$ achieve higher average scores (from 2.56% to 4.09%) compared to DPO. This indicates the effectiveness of SGDPO. Moreover, *Pilot*$_d$ attains a higher average score (1.50%) than *Pilot*$_s$. This is expected as different random indices introduce an element of randomness into the learning space corresponding to superior performance as explained in Section 3.4.

r_1	r_2	Score$_1$	Score$_2$	Score$_{\text{avg}}$
0.9	0.5	7.72	6.50	7.11
0.9	0.6	**8.01**	**6.95**	**7.48**
0.9	0.7	7.84	6.89	7.37
0.9	0.8	7.84	6.68	7.26
0.9	0.9	7.98	6.90	7.44

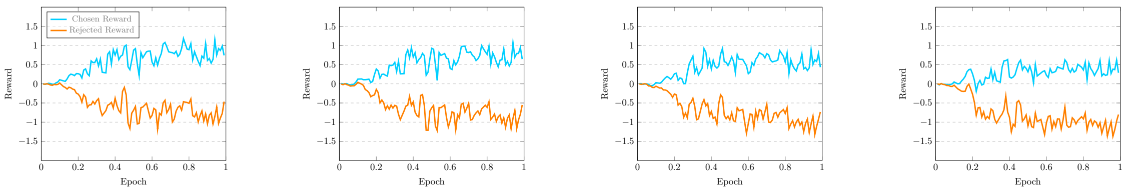
Table 3: MT-Bench Results across different r_1 and r_2 on Llama-3.1 base 8B model.

4.3.2 The *Pilot* Term

While r_1 and r_2 are critical to *pilot*, we conduct experimentation to evaluate the influence of r_1 and r_2 on the performance of SGDPO, including the robustness and the reward patterns.

Robustness. While *pilot* exploits r_1 and r_2 to regulate the lengths of the token sequences, we carry out an experimentation with diverse r_1 and r_2 ($r_1 = r_2 = r$) so as to verify the corresponding performance and robustness of SGDPO. As shown in Figure 5, SGDPO significantly outperforms DPO in all settings, achieving notably higher scores in the first turn (from 4.25% to 9.19%), second turn (from 1.56% to 10.14%), and on average (from 3.06% to 8.61%). SGDPO achieves its lowest value at $r = 1$, yet it still exhibits a relative improvement of 3.06% over DPO. The length of the generated response tokens remains comparable to that of DPO. These experimental results show the robustness of SGDPO across varying r_1 and r_2.

Reward Patterns. As shown in Figure 3, when r_1 and r_2 range from 0.6 to 0.9, both the average Convergence Chosen Reward (CCR) and the average Convergence Reject Reward (CRR) over the last 80 iterations of fine-tuning decrease. A more significant CRR corresponds to a modest decrease in rejected rewards, which is in line with G1 explained in Section 3.3. Correspondingly, as shown in Figure 5, the average score is negatively correlated with r (from 0.6 to 1.0), with the exception of a fluctuation occurring at $r = 0.9$. This fluctuation may be due to the randomness in subsequence construction. As a consequence, in our experimentation, we take the best values of r to achieve excellent performance (see experimental setting details of r in Appendix).



(a) CCR = 0.7542 CRR = -0.8325 (b) CCR = 0.6816 CRR = -0.9011 (c) CCR = 0.5267 CRR = -1.0337 (d) CCR = 0.3569 CRR = -1.0842

Figure 3: Training reward curves for the Llama-3.1 base 8B model using the SGDPO method: (a) $r_1 = 0.6$ and $r_2 = 0.6$. (b) $r_1 = 0.7$ and $r_2 = 0.7$. (c) $r_1 = 0.8$ and $r_2 = 0.8$. (d) $r_1 = 0.9$ and $r_2 = 0.9$. "CCR" represents the average (last 80 iterations) convergence chosen reward and "CRR" represents the average (last 80 iterations) convergence reject reward.

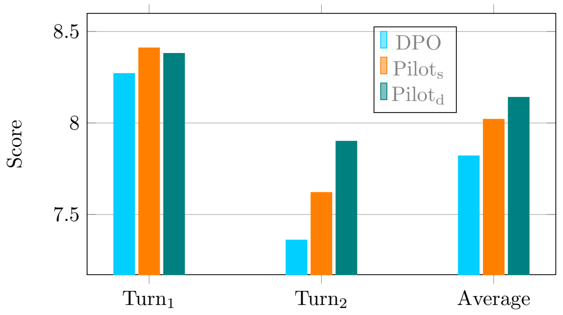


Figure 4: MT-Bench Results across different model configurations, using Llama-3.1 instruct 8B as the base model.

Method	Score$_1$	Score$_2$	Score$_{avg}$
DPO	7.29	6.41	6.85
ORPO	7.40	6.04	6.78
SGDPO	**8.01**	**6.95**	**7.48**

Table 4: MT-Bench Results across different methods on Llama-3.1 base 8B model.

Different r_1 and r_2. In previous experiments, we set $r_1 = r_2$ to evaluate model performance. To further investigate the effectiveness of SGDPO, we conduct additional experiments with different values of r_1 and r_2. As shown in Table 3, varying these parameters leads to further improvements in performance. Specifically, setting $r_1 = 0.9$ and $r_2 = 0.6$ achieves an average score of 7.48, outperforming the baseline configuration ($r_1 = r_2 = 0.9$, score = 7.44). This result also represents a significant improvement over DPO, with a relative gain of 9.19%.

Compared with ORPO. ORPO (Hong et al., 2024) presents a novel approach to preference optimization by proposing a unified odds ratio-based framework that does not rely on a separate reference model. This innovative method effectively integrates preference learning into a single training stage, thereby removing the need for an additional alignment step and significantly streamlining the overall optimization process.

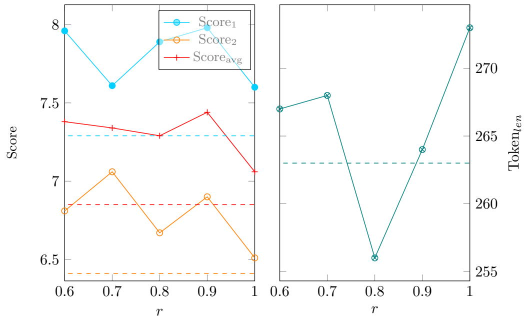


Figure 5: MT-Bench results of SGDPO across various configurations, using Llama-3.1 base 8B as the base model. The dashed lines represent the score and the token length of DPO.

We conduct an ablation study to compare SGDPO, DPO, and ORPO. As shown in Table 4, SGDPO outperforms both DPO and ORPO by a large margin, which demonstrates the effectiveness of SGDPO.

5 Conclusions

In this paper, we present a novel self-guided direct preference optimization algorithm, i.e., SGDPO, for aligning LLMs with human preferences. SGDPO incorporates a *pilot* term in the objective function in order to guide the gradient updates of the rewards during training. We provide a detailed theoretical explanation of SGDPO. Furthermore, extensive experimental results across various model settings and benchmarks demonstrate the significant advantages (up to 9.19% higher score) of SGDPO.

Limitations

SGDPO includes the resampling of a sub-sequence from the logits of the output layer, which introduces extra computational steps. Nevertheless, as demonstrated in Table 8 within the Appendix, this results in a minor increase (up to 0.4%) in computational overhead.

While SGDPO exploits public centralized preference datasets to fine-tune models in order to align LLMs with human values, the datasets may contain unhelpful or misleading preferred information leading to unexpected responses. SGDPO may be subject to this potential drawback. In addition, the datasets may be distributed in diverse data centers or edge devices (Chen et al., 2025; Liu et al., 2024b, 2022a, 2015), which may restrict the application of SGDPO. In the future, we plan to investigate the adaptation of SGDPO into a broader setting, e.g., federated learning (Liu et al., 2024e,d; Jia et al., 2024; Liu et al., 2024a,c,c; Che et al., 2023; Liu et al., 2023b, 2022b; Zhang et al., 2022; Zhou et al., 2022) and distributed machine learning (Liu et al., 2023a).

References

AI@Meta. 2024. Introducing llama 3.1: Our most capable models to date.

Anthropic. 2024. Claude 3.5 sonnet.

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Remi Munos. 2023. A general theoretical paradigm to understand learning from human preferences. *arXiv preprint arXiv:2310.12036*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom B. Brown, Jack Clark, Sam McCandlish, Chris Olah, Benjamin Mann, and Jared Kaplan. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, Huazuo Gao, Kaige Gao, Wenjun Gao, Ruiqi Ge, Kang Guan, Daya Guo, Jianzhong Guo, Guangbo Hao, Zhewen Hao, Ying He, Wenjie Hu, Panpan Huang, Erhang Li, Guowei Li, Jiashi Li, Yao Li, Y. K. Li, Wenfeng Liang, Fangyun Lin, Alex X. Liu, Bo Liu, Wen Liu, Xiaodong Liu, Xin Liu, Yiyuan Liu, Haoyu Lu, Shanghao Lu, Fuli Luo, Shirong Ma, Xiaotao Nie, Tian Pei, Yishi Piao, Junjie Qiu, Hui Qu, Tongzheng Ren, Zehui Ren, Chong Ruan, Zhangli Sha, Zhihong Shao, Junxiao Song, Xuecheng Su, Jingxiang Sun, Yaofeng Sun, Minghui Tang, Bingxuan Wang, Peiyi Wang, Shiyu Wang, Yaohui Wang, Yongji Wang, Tong Wu, Y. Wu, Xin Xie, Zhenda Xie, Ziwei Xie, Yiliang Xiong, Hanwei Xu, R. X. Xu, Yanhong Xu, Dejian Yang, Yuxiang You, Shuiping Yu, Xingkai Yu, B. Zhang, Haowei Zhang, Lecong Zhang, Liyue Zhang, Mingchuan Zhang, Minghua Zhang, Wentao Zhang, Yichao Zhang, Chenggang Zhao, Yao Zhao, Shangyan Zhou, Shunfeng Zhou, Qihao Zhu, and Yuheng Zou. 2024. Deepseek LLM: scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. In *AAAI*.

Ralph Allan Bradley and Milton E. Terry. 1952a. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*.

Ralph Allan Bradley and Milton E Terry. 1952b. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*.

Tianshi Che, Ji Liu, Yang Zhou, Jiaxiang Ren, Jiwen Zhou, Victor S. Sheng, Huaiyu Dai, and Dejing Dou. 2023. Federated learning of large language models with parameter-efficient prompt tuning and adaptive optimization. In *Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7871–7888.

Chunlu Chen, Ji Liu, Haowen Tan, Xingjian Li, Kevin I-Kai Wang, Peng Li, Kouichi Sakurai, and Dejing Dou. 2025. Trustworthy federated learning: privacy, security, and beyond. *Knowledge and Information Systems*, 67(3):2321–2356.

Huayu Chen, Guande He, Lifan Yuan, Ganqu Cui, Hang Su, and Jun Zhu. 2024. Noise contrastive alignment of language models with explicit rewards. *arXiv preprint arXiv:2402.05369*.

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 arXiv:1803.05457*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, 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. 2023. Ultrafeedback: Boosting language models with high-quality feedback. *arXiv preprint arXiv:2310.01377*.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233*.

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*.

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Alpacafarm: A simulation framework for methods that learn from human feedback. *arXiv preprint arXiv:2305.14387*.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. 2020. Implementation matters in deep rl: A case study on ppo and trpo. In *ICLR*.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*.

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*.

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 *AISTATS*.

Google. 2024. Our next-generation model: Gemini 1.5.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Xiaodong Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Jiwoo Hong, Noah Lee, and James Thorne. 2024. ORPO: Monolithic preference optimization without reference model. In *EMNLP*.

Zhenyu Hou, Yiin Niu, Zhengxiao Du, Xiaohan Zhang, Xiao Liu, Aohan Zeng, Qinkai Zheng, Minlie Huang, Hongning Wang, Jie Tang, and Yuxiao Dong. 2024. Chatglm-rlhf: Practices of aligning large language models with human feedback. *arXiv preprint arXiv:2404.00934*.

Andrew Ilyas, Logan Engstrom, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. 2020. A closer look at deep policy gradients. In *ICLR*.

Juncheng Jia, Ji Liu, Chendi Zhou, Hao Tian, Mianxiong Dong, and Dejing Dou. 2024. Efficient asynchronous federated learning with sparsification and quantization. *Concurrency and Computation: Practice and Experience*, 36(9):e8002.

Seungjae Jung, Gunsoo Han, Daniel Wontae Nam, and Kyoung-Woon On. 2024a. Binary classifier optimization for large language model alignment. *arXiv preprint arXiv:2404.04656*.

Seungjae Jung, Gunsoo Han, Daniel Wontae Nam, and Kyoung-Woon On. 2024b. Binary classifier optimization for large language model alignment. *arXiv preprint arXiv:2404.04656*.

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 *ACL*.

Ji Liu, Tianshi Che, Yang Zhou, Ruoming Jin, Huaiyu Dai, Dejing Dou, and Patrick Valduriez. 2024a. Aedfl: efficient asynchronous decentralized federated learning with heterogeneous devices. In *SIAM Int. Conf. on Data Mining (SDM)*, pages 833–841. SIAM.

Ji Liu, Chunlu Chen, Yu Li, Lin Sun, Yulun Song, Jingbo Zhou, Bo Jing, and Dejing Dou. 2024b. Enhancing trust and privacy in distributed networks: a comprehensive survey on blockchain-based federated learning. *Knowledge and Information Systems*, 66(8):4377–4403.

Ji Liu, Jizhou Huang, Yang Zhou, Xuhong Li, Shilei Ji, Haoyi Xiong, and Dejing Dou. 2022a. From distributed machine learning to federated learning: A survey. *Knowledge and Information Systems*, 64(4):885–917.

Ji Liu, Juncheng Jia, Tianshi Che, Chao Huo, Jiaxiang Ren, Yang Zhou, Huaiyu Dai, and Dejing Dou. 2024c. Fedasmu: Efficient asynchronous federated learning with dynamic staleness-aware model update. In *AAAI Conference on Artificial Intelligence*, volume 38, pages 13900–13908.

Ji Liu, Juncheng Jia, Beichen Ma, Chendi Zhou, Jingbo Zhou, Yang Zhou, Huaiyu Dai, and Dejing Dou. 2022b. Multi-job intelligent scheduling with cross-device federated learning. *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, 34(2):535–551.

Ji Liu, Juncheng Jia, Hong Zhang, Yuhui Yun, Leye Wang, Yang Zhou, Huaiyu Dai, and Dejing Dou. 2024d. Efficient federated learning using dynamic update and adaptive pruning with momentum on shared server data. *ACM Transactions on Intelligent Systems and Technology*, 15(6):1–28.

Ji Liu, Esther Pacitti, Patrick Valduriez, and Marta Mattoso. 2015. A survey of data-intensive scientific workflow management. *Journal of Grid Computing*, 13:457–493.

Ji Liu, Jiaxiang Ren, Ruoming Jin, Zijie Zhang, Yang Zhou, Patrick Valduriez, and Dejing Dou. 2024e. Fisher information-based efficient curriculum federated learning with large language models. In *Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, pages 10497–10523.

Ji Liu, Zhihua Wu, Danlei Feng, Minxu Zhang, Xinxuan Wu, Xuefeng Yao, Dianhai Yu, Yanjun Ma, Feng Zhao, and Dejing Dou. 2023a. Heterps: Distributed deep learning with reinforcement learning based scheduling in heterogeneous environments. *Future Generation Computer Systems*, 148:106–117.

Ji Liu, Xuehai Zhou, Lei Mo, Shilei Ji, Yuan Liao, Zheng Li, Qin Gu, and Dejing Dou. 2023b. Distributed and deep vertical federated learning with big data. *Concurrency and Computation: Practice and Experience*, 35(21):e7697.

Jiaheng Liu, Dawei Zhu, Zhiqi Bai, Yancheng He, Huanxuan Liao, Haoran Que, Zekun Wang, Chenchen Zhang, Ge Zhang, Jiebin Zhang, et al. 2025. A comprehensive survey on long context language modeling. *arXiv preprint arXiv:2503.17407*.

Wei Liu, Yang Bai, Chengcheng Han, Rongxiang Weng, Jun Xu, Xuezhi Cao, Jingang Wang, and Xunliang Cai. 2024f. Length desensitization in direct 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. *arXiv preprint arXiv:2406.10957*.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. In *NeurIPS*.

OpenAI. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

OpenAI. 2024. Hello gpt-4o.

Long Ouyang, Jeff 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 Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*.

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 arXiv:2402.13228*.

Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. 2024. Disentangling length from quality in direct preference optimization. In *ACL*.

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 *NeurIPS*.

Noam Razin, Sadhika Malladi, Adithya Bhaskar, Danqi Chen, Sanjeev Arora, and Boris Hanin. 2025. Unintentional unalignment: Likelihood displacement in direct preference optimization. In *ICLR*.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2020. Learning to summarize from human feedback. In *NeurIPS*.

Teng Xiao, Yige Yuan, Zhengyu Chen, Mingxiao Li, Shangsong Liang, Zhaochun Ren, and Vasant G Honavar. 2025. Simper: A minimalist approach to preference alignment without hyperparameters. In *ICLR*.

Teng Xiao, Yige Yuan, Huaisheng Zhu, Mingxiao Li, and Vasant G Honavar. 2024. Cal-dpo: Calibrated direct preference optimization for language model alignment. In *NeurIPS*.

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 *ICML*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Yongcheng Zeng, Guoqing Liu, Weiyu Ma, Ning Yang, Haifeng Zhang, and Jun Wang. 2024. Token-level direct preference optimization. *arXiv preprint arXiv:2404.11999*.

Hong Zhang, Ji Liu, Juncheng Jia, Yang Zhou, Huaiyu Dai, and Dejing Dou. 2022. Fedduap: Federated learning with dynamic update and adaptive pruning using shared data on the server. In *Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 2776–2782.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*.

Chendi Zhou, Ji Liu, Juncheng Jia, Jingbo Zhou, Yang Zhou, Huaiyu Dai, and Dejing Dou. 2022. Efficient device scheduling with multi-job federated learning. In *AAAI Conf. on Artificial Intelligence*, volume 36, pages 9971–9979.

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 arXiv:2311.07911*.

Wenqiao Zhu, Lulu Wang, and Jun Wu. 2025. Addressing cold-start problem in click-through rate prediction via supervised diffusion modeling. In *AAAI*.

Wenqiao Zhu, Chao Xu, Lulu Wang, and Jun Wu. 2024. Psc: Extending context window of large language models via phase shift calibration. In *EMNLP*.

A Appendix

A.1 Proof of 1, 2, and 3

Theorem 1. *The partial derivatives of l_{pilot} with respect to $\mathcal{X}_1$ and $\mathcal{X}_2$ are given by:*

$$\frac{\partial l_{pilot}}{\partial \mathcal{X}_1} = \frac{\beta \mathcal{Y}_2^\beta}{\mathcal{X}_1(\mathcal{X}_1^\beta + \mathcal{Y}_2^\beta)} \tag{1}$$

$$\frac{\partial l_{pilot}}{\partial \mathcal{X}_2} = -\frac{\beta \mathcal{X}_2^{\beta-1}}{\mathcal{Y}_1^\beta + \mathcal{X}_2^\beta} \tag{2}$$

Proof. By variable substitution, we have:

$$l_{pilot}(\pi_\theta, \pi_{\text{ref}}) = \log\left(\frac{\mathcal{X}_1^\beta}{\mathcal{X}_1^\beta + \mathcal{Y}_2^\beta}\right) + \log\left(\frac{\mathcal{Y}_1^\beta}{\mathcal{Y}_1^\beta + \mathcal{X}_2^\beta}\right) \tag{3}$$

For $\frac{\partial l_{\text{pilot}}}{\partial \mathcal{X}_1}$,

$$\begin{aligned}\frac{\partial l_{\text{pilot}}}{\partial \mathcal{X}_1} &= \frac{\mathcal{X}_1^\beta + \mathcal{Y}_2^\beta}{\mathcal{X}_1^\beta}\left(\frac{\beta \mathcal{X}_1^{\beta-1}}{\mathcal{X}_1^\beta + \mathcal{Y}_2^\beta} - \frac{\beta \mathcal{X}_1^{2\beta-1}}{(\mathcal{X}_1^\beta + \mathcal{Y}_2^\beta)^2}\right) \\ &= \frac{\beta \mathcal{Y}_2^\beta}{\mathcal{X}_1(\mathcal{X}_1^\beta + \mathcal{Y}_2^\beta)} \qquad (4)\end{aligned}$$

For $\frac{\partial l_{\text{pilot}}}{\partial \mathcal{X}_2}$,

$$\begin{aligned}\frac{\partial l_{\text{pilot}}}{\partial \mathcal{X}_2} &= \frac{\mathcal{Y}_1^\beta + \mathcal{X}_2^\beta}{\mathcal{Y}_1^\beta}\frac{-\mathcal{Y}_1^\beta \beta \mathcal{X}_2^{\beta-1}}{(\mathcal{Y}_1^\beta + \mathcal{X}_2^\beta)^2} \\ &= -\frac{\beta \mathcal{X}_2^{\beta-1}}{\mathcal{Y}_1^\beta + \mathcal{X}_2^\beta} \qquad (5)\end{aligned}$$

□

Theorem 2. *The partial derivative $|\frac{\partial l_{pilot}}{\partial \mathcal{X}_1}|$ increases as $\mathcal{Y}_2$ increases, while the partial derivative $|\frac{\partial l_{pilot}}{\partial \mathcal{X}_2}|$ descreases as $\mathcal{Y}_1$ increases.*

Proof. For $|\frac{\partial l_{\text{pilot}}}{\partial \mathcal{X}_1}|$, we have

$$\begin{aligned}\frac{\partial |\frac{\partial l_{\text{pilot}}}{\partial \mathcal{X}_1}|}{\partial \mathcal{Y}_2} &= \frac{\beta^2 \mathcal{Y}^{\beta-1} \mathcal{X}_1^\beta}{\mathcal{X}_1(\mathcal{Y}_2^\beta + \mathcal{X}_1^\beta)^2} \qquad (6) \\ &> 0 \qquad (7)\end{aligned}$$

For $|\frac{\partial l_{\text{pilot}}}{\partial \mathcal{X}_2}|$, we have

$$\begin{aligned}\frac{\partial |\frac{\partial l_{\text{pilot}}}{\partial \mathcal{X}_2}|}{\partial \mathcal{Y}_1} &= -\frac{\beta^2 \mathcal{X}_2^{\beta-1} \mathcal{Y}_1^{\beta-1}}{(\mathcal{Y}_1^\beta + \mathcal{X}_2^\beta)^2} \qquad (8) \\ &< 0 \qquad (9)\end{aligned}$$

□

Theorem 3. *Let $\pi_{pilot} = \pi_\theta$ and $z = \frac{\mathcal{Y}_1}{\mathcal{Y}_2}$, for each pairwise preference instance $(x, y_w, y_l) \in \mathcal{D}$, the ratio of the increase in the probability of a human-preferred response to the decrease in the probability of a human-dispreferred response is given by:*

$$\left|\frac{\partial l_{pilot}}{\partial \mathcal{X}_1} / \frac{\partial l_{pilot}}{\partial \mathcal{X}_2}\right| = \frac{\mathcal{X}_2}{\mathcal{X}_1} \cdot f(z), \tag{10}$$

where

$$f(z) = \frac{1}{p_2^\beta}\frac{z^\beta + p_2^\beta}{p_1^\beta z^\beta + 1} \tag{11}$$

is a monotonic function of z. When $p_1 p_2 < 1$, the function $f(z)$ is increasing. Conversely, if $p_1 p_2 > 1$, the function $f(z)$ is decreasing. Furthermore, $f(z) > 1$ if $p_1 p_2 < 1$.

Proof.

$$\begin{aligned}\left|\frac{\partial \mathcal{L}_{\text{pilot}}}{\partial \mathcal{X}_1} / \frac{\partial \mathcal{L}_{\text{pilot}}}{\partial \mathcal{X}_2}\right| &= \frac{\mathcal{Y}_2}{\mathcal{X}_1}\frac{\mathcal{Y}_2^{\beta-1}}{\mathcal{X}_2^{\beta-1}}\frac{\mathcal{Y}_1^\beta + \mathcal{X}_2^\beta}{\mathcal{X}_1^\beta + \mathcal{Y}_2^\beta} \qquad (12) \\ &= \frac{\mathcal{X}_2}{\mathcal{X}_1}\frac{\mathcal{Y}_2^\beta}{\mathcal{X}_2^\beta}\frac{\mathcal{Y}_1^\beta + \mathcal{X}_2^\beta}{\mathcal{X}_1^\beta + \mathcal{Y}_2^\beta} \qquad (13)\end{aligned}$$

Let $\mathcal{X}_2 = p_2 \mathcal{Y}_2, \mathcal{X}_1 = p_1 \mathcal{Y}_1$, and $z = \frac{\mathcal{Y}_1}{\mathcal{Y}_2}$, we then have

$$\begin{aligned}f(z) &= \frac{\mathcal{Y}_2^\beta}{\mathcal{X}_2^\beta}\frac{\mathcal{Y}_1^\beta + \mathcal{X}_2^\beta}{\mathcal{X}_1^\beta + \mathcal{Y}_2^\beta} \\ &= \frac{\mathcal{Y}_2^\beta}{(p_2 \mathcal{Y}_2)^\beta}\frac{\mathcal{Y}_1^\beta + (p_2 \mathcal{Y}_2)^\beta}{(p_1 \mathcal{Y}_1)^\beta + \mathcal{Y}_2^\beta} \\ &= \frac{1}{p_2^\beta}\frac{z^\beta + p_2^\beta}{p_1^\beta z^\beta + 1} \qquad (14)\end{aligned}$$

The derivative of $f(z)$ with respect to z is

$$\begin{aligned}\frac{\partial f(z)}{\partial z} &\propto \beta z^{\beta-1}(p_1^\beta z^\beta + 1) - (z^\beta + p_2^\beta) p_1^\beta \beta z^{\beta-1} \\ &= \beta z^{\beta-1} - \beta (p_1 p_2)^\beta z^{\beta-1} \\ &= \beta\left(1 - (p_1 p_2)^\beta\right) z^{\beta-1} \qquad (15)\end{aligned}$$

Since $z = \frac{\mathcal{Y}_1}{\mathcal{Y}_2} > 0$, whether $\frac{\partial f(z)}{\partial z} > 0$ or $\frac{\partial f(z)}{\partial z} < 0$ is contingent on the value of $p_1 p_2$. Therefore, if $p_1 p_2 < 1$, the function $f(z)$ is increasing. Conversely, if $p_1 p_2 > 1$, the function $f(z)$ is decreasing. □

A.2 Experimental Setup

To ensure a fair comparison among different methods, we employ the same general settings for all baselines, which are detailed in Table 6. Additionally, we set $\beta = 0.1$ for all baselines. For the proposed SGDPO method, we set $r_1 = r_2$ by default and performed a grid search over the range $\{0.6, 0.7, \cdots, 1.0\}$. Table 5 shows the parameters we select. We carry out our experiments on 4 A800-80G GPUs.

As a large-scale, finely detailed, and diverse dataset, UltraFeedback dataset (Cui et al., 2023)

Model	r_1, r_2
Llama-3.1 instruct 8B	$r_1 = 1.0$, $r_2 = 1.0$
Llama-3.1 Base 8B	$r_1 = 0.9$, $r_2 = 0.9$
Qwen-2 instruct 7B	$r_1 = 0.9$, $r_2 = 0.9$
Qwen-2 base 7B	$r_1 = 0.6$, $r_2 = 0.6$

Table 5: The hyper-parameters we used for SGDPO in the experiments reported in Table 1

Phase	LR	BS	Epoch	LS	WP
SFT	2e-5	128	3	cosine	0.1
PO	5e-7	128	1	cosine	0.1

Table 6: The general training settings for the Supervised Fine - Tuning (SFT) phase and Preference Optimization (PO) phase include Learning Rate (LR), Batch Size (BS), Epoch, Learning Rate Schedule (LS), and Warmup Phase (WP).

comprises approximately 64,000 prompts sourced from a wide array of origins. MT-Bench consists of a multi-turn question set with 80 questions designed to evaluate the capabilities of a model in multi-turn conversation and instruction-following. In our experimentation, we utilize a single-answer grading mode, where GPT-4 (OpenAI, 2023) assigns a score out of 10 for each turn. We report the average score per turn across our experiments.

B Complexity

SGDPO entails a novel technique where we resample subsequences from the probability distributions (logits) generated by the output layer. This process introduces supplementary computational stages into the workflow. Despite this added complexity, as detailed in Table 8, the resultant increase in computational overhead remains modest (up to 0.4%) additional computational time.

C More Experiments

We also employ the AlpacaEval-2 (Li et al., 2023; Dubois et al., 2024) benchmark for evaluation. AlpacaEval-2 operates on a fixed set of 805 instructions, for which both the base model and the evaluated model generate responses. A GPT-based model then compares these responses to determine the win rate. In our experiments, we report both the length-controlled win rate and the raw win rate. We utilize the *weighted_alpaca_eval_gpt4_turbo* configuration recommended by the AlpacaEval-2 library (Dubois et al., 2024) for this evaluation. We report the results in Table 7.

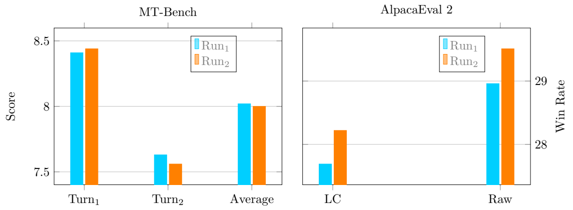


Figure 6: Performance Metrics of Various Runs on MT-Bench and AlpacaEval-2.

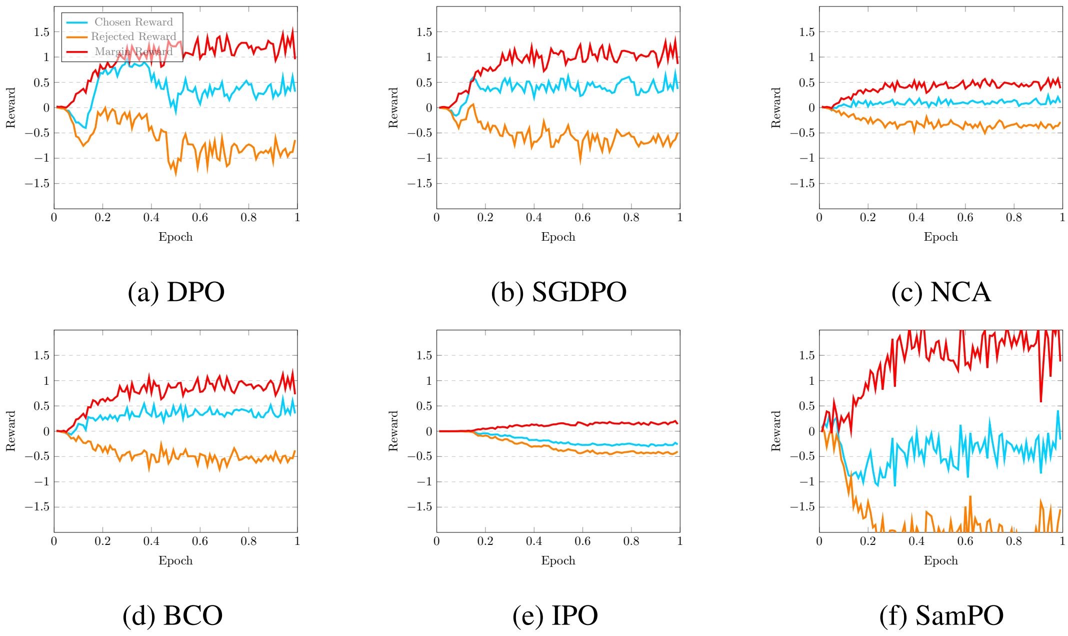
From the experimental results, we can observe that SGDPO significantly outperforms the baselines in the LC win rate metric (up to 5.14%) with Llama-3.1 instruct 8B. However, unlike the experiments on MT-Bench, SGDPO does not surpass the baselines on Qwen-2 instruct 7B model. This indicates the effectiveness of alignment optimization might be benchmark-dependent. Conducting a rigorous evaluation of large language models remains a research direction of significant importance.

D Training Reward Curves

As discussed in Section 3.4, adjusting the values of r_1 and r_2 can affect the optimization process, resulting in different reward curve shapes. In Figure 8, we present the full training curves for SGDPO and DPO. The results show that setting r_1 and r_2 to smaller values can lead to an increase in the magnitude of the reward values at the end of the fine-tuning stage. We also present the training reward curves of the baselines in Figure 7.

E Variance

In this paper, we carry out extensive experiments using both the MT-Bench and AlpacaEval-2 frameworks. Both MT-Bench and AlpacaEval-2 utilize GPT for evaluating responses, we investigate whether there are significant discrepancies in the assessments of GPT with identical content across different calls. To explore this, we conducted a test by querying GPT twice with the same response content and present our findings in Figure 6. The experimental results indicate that while MT-Bench yields relatively consistent outcomes with lower variance, AlpacaEval-2 demonstrates a notably higher variance under similar conditions.



(a) DPO (b) SGDPO (c) NCA

(d) BCO (e) IPO (f) SamPO

Figure 7: Training reward curves for the Llama-3.1 instruct 8B model using various alignment methods.

Methods	Llama-3.1 instruct 8B			Qwen-2 instruct 7B		
	LC win rate	Raw win rate	Token$_{\text{len}}$	LC win rate	Raw win rate	Token$_{\text{len}}$
SFT	26.84	27.77	459	20.98	22.20	418
DPO (Rafailov et al., 2023)	27.53	28.35	438	24.26	24.50	414
NCA (Chen et al., 2024)	26.33	27.77	441	21.94	21.75	409
BCO (Jung et al., 2024b)	28.03	**29.32**	435	23.76	23.95	411
IPO (Gheshlaghi Azar et al., 2024)	27.07	25.82	459	**29.03**	25.68	411
SamPO (Lu et al., 2024)	27.45	27.69	443	24.57	**26.60**	426
SGDPO	**28.22**	28.96	444	23.89	24.86	419

Table 7: AlpacaEval-2 Results across different model configurations. Token$_{\text{len}}$ indicates the average length of output tokens for each method.

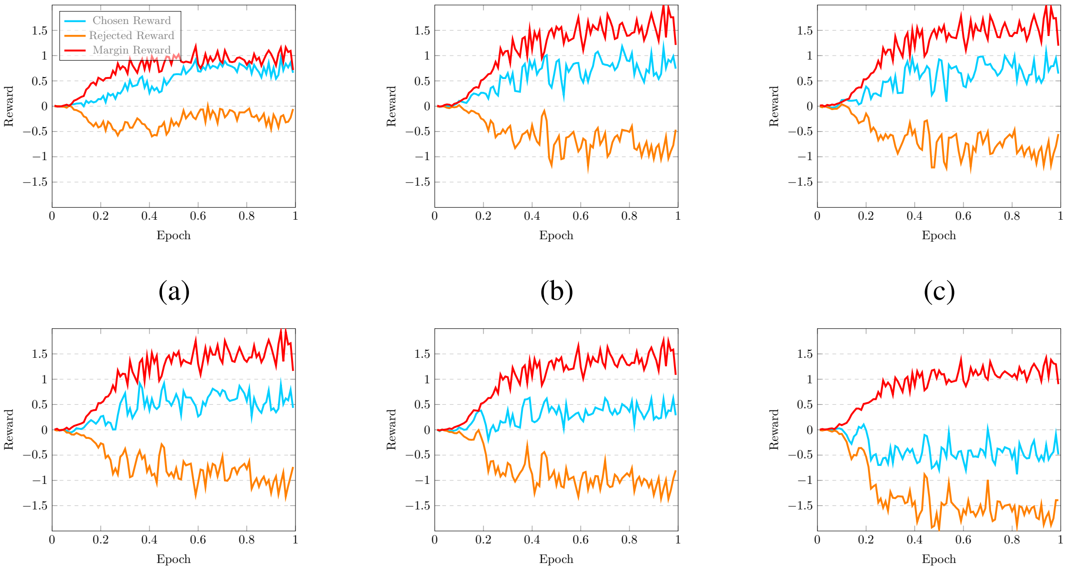
Method	Training Time
DPO	6h22m22s
SGDPO	6h24m07s

Table 8: Training time cost of DPO and SGDPO.

F Future Work

As discussed in the Limitations section, SGDPO introduces additional computational steps. To address this, we aim to design a novel architecture for SGDPO that reduces the associated computational overhead. We also plan to evaluate our method in long-context scenarios (Liu et al., 2025; Zhu et al., 2024) and recommendation systems (Zhu et al., 2025), as recommendations are inherently driven by user preferences.

Additionally, we intend to explore the applicability of SGDPO in broader settings, such as learning with non-Independent and Identically Distributed (non-IID) data under federated learning frameworks. We also plan to investigate the use of diverse models or enhanced architectures within the policy framework—specifically, the *pilot* model in SGDPO—to further improve alignment performance. Finally, we aim to develop new self-guidance mechanisms for preference optimization and explore how SGDPO can be leveraged to enhance the reasoning capabilities of large language models (LLMs).



(a)

(b)

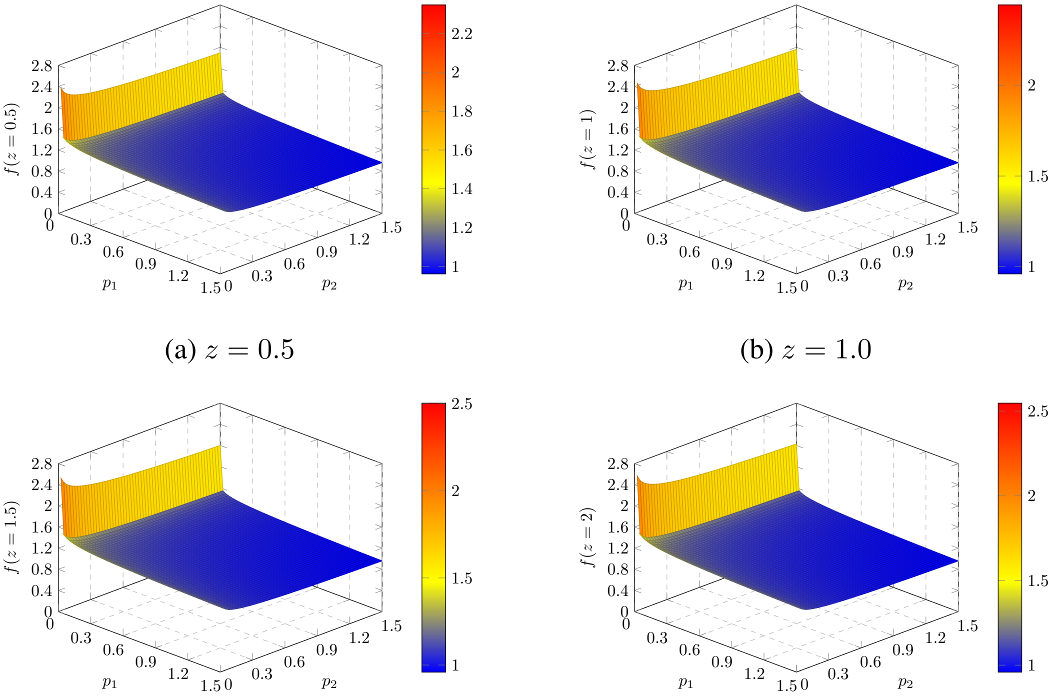
(c)

(d)

(e)

(f)

Figure 8: Training reward curves for the Llama-3.1 base 8B model using the DPO and SGDPO methods: (a) DPO. (b) SGDPO with $r_1 = 0.6$ and $r_2 = 0.6$. (c) SGDPO with $r_1 = 0.7$ and $r_2 = 0.7$. (d) SGDPO with $r_1 = 0.8$ and $r_2 = 0.8$. (e) SGDPO with $r_1 = 0.9$ and $r_2 = 0.9$. (f) SGDPO with $r_1 = 1.0$ and $r_2 = 1.0$.



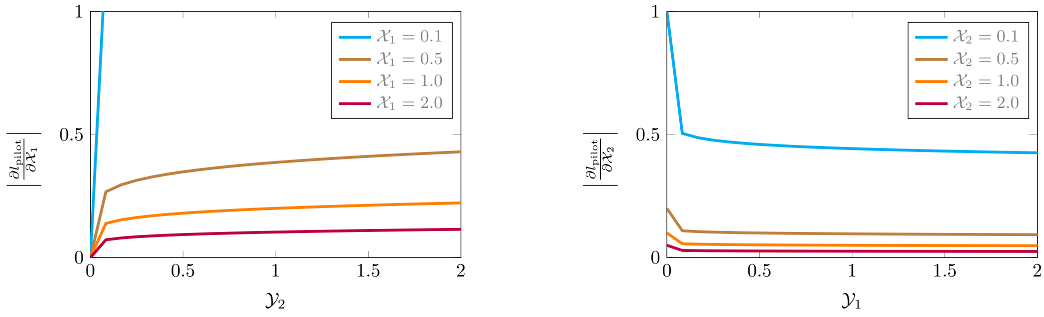
(a) $z = 0.5$

(b) $z = 1.0$

$z = 1.5$

(d) $z = 2.0$

Figure 9: Visual representation of the function $f(z)$ landscape.



(a)

(b)

Figure 10: Visual representation of the functions $\left|\frac{\partial l_{\text{pilot}}}{\partial \mathcal{X}_1}\right|$ and $\left|\frac{\partial l_{\text{pilot}}}{\partial \mathcal{X}_2}\right|$ at selected fixed points.