Self-supervised Preference Optimization: Enhance Your Language Model with Preference Degree Awareness

Jian Li[1,*], Haojing Huang[1,*], Yujia Zhang[1,†], Pengfei Xu[1], Xi Chen[2†], Rui Song[3], Lida Shi[4], Jingwen Wang[3], Hao Xu[3,4]

[1]AI Technology Center of OVB, Tencent, [2]Platform and Content Group, Tencent,
[3]College of Computer Science and Technology, Jilin University,
[4]School of Artificial Intelligence, Jilin University

{loucasli,waterrhuang,yujiazhang,luciferxu,jasonxchen}@tencent.com, {songrui,xuhao}@jlu.edu.cn,
{shild21,wjw22}@mails.jlu.edu.cn

Abstract

Recently, there has been significant interest in replacing the reward model in Reinforcement Learning with Human Feedback (RLHF) methods for Large Language Models (LLMs), such as Direct Preference Optimization (DPO) and its variants. These approaches commonly use a binary cross-entropy mechanism on pairwise samples, i.e., minimizing and maximizing the loss based on preferred or dis-preferred responses, respectively. However, while this training strategy omits the reward model, it also overlooks the varying preference degrees within different responses. We hypothesize that this is a key factor hindering LLMs from sufficiently understanding human preferences. To address this problem, we propose a novel Self-supervised Preference Optimization (SPO) framework, which constructs a self-supervised preference degree loss combined with the alignment loss, thereby helping LLMs improve their ability to understand the degree of preference. Extensive experiments are conducted on two widely used datasets of different tasks. The results demonstrate that SPO can be seamlessly integrated with existing preference optimization methods and significantly boost their performance to achieve state-of-the-art performance. We also conduct detailed analyses to offer comprehensive insights into SPO, which verifies its effectiveness. The code is available at https://github.com/lijian16/SPO.

1 Introduction

The alignment of Large Language Models (LLMs) with human preferences is paramount, as it ensures that the outputs of LLMs are congruent with human values and ethical standards (Böhm et al., 2019; Perez et al., 2019; Ziegler et al., 2019). Through meticulous tuning and ongoing learning of human preferences, LLMs can more accurately meet user needs while avoiding the generation of harmful or biased content (Stiennon et al., 2020b; Lee et al., 2024). Effective preference alignment not only enhances the applicability and safety of LLMs but also constitutes a critical step towards the responsible utilization of artificial intelligence.

To achieve human preference alignment of LLMs, a variety of methods have been developed. One prominent approach is Reinforcement Learning from Human Feedback (RLHF) (Stiennon et al., 2020b; Bai et al., 2022), such as Proximal Policy Optimization (PPO) (Schulman et al., 2017), REINFORCE (Williams, 1992) and their variants (Ramamurthy et al., 2023), which involve training reward models to optimize for objectives that are iteratively refined based on human feedback. However, these methods introduce increased complexity into the training process, involving training multiple models and sampling from the LLM in the loop of training (Ethayarajh et al., 2024; Yuan et al., 2024). To streamline this process, recent works have proposed alternative solutions to reinforcement learning (Liu et al., 2023a; Zhao et al., 2023; Ethayarajh et al., 2024; Azar et al., 2023). DPO (Rafailov et al., 2023) and its variants (Wang et al., 2023; Song et al., 2024; Ethayarajh et al., 2024; Azar et al., 2024; Amini et al., 2024; Meng et al., 2024; Yu et al., 2024) directly leverage pairwise responses to imbue the model with preference knowledge without a reward function. These methods achieve preference alignment by minimizing or maximizing the loss between each token in the language model's output and the tokens that are either preferred or not preferred. However, this training strategy overlooks a crucial aspect of a reward model: its ability to differentiate between varying degrees of human preferences in responses. We hypothesize that this is a key factor that prevents LLMs from fully understanding human preferences in those RLHF methods without a reward model.

To address this issue, we propose a novel Self-

[*]These authors contributed equally to this work.
[†]Corresponding author

Findings of the Association for Computational Linguistics: EACL 2024, pages 14452–14466

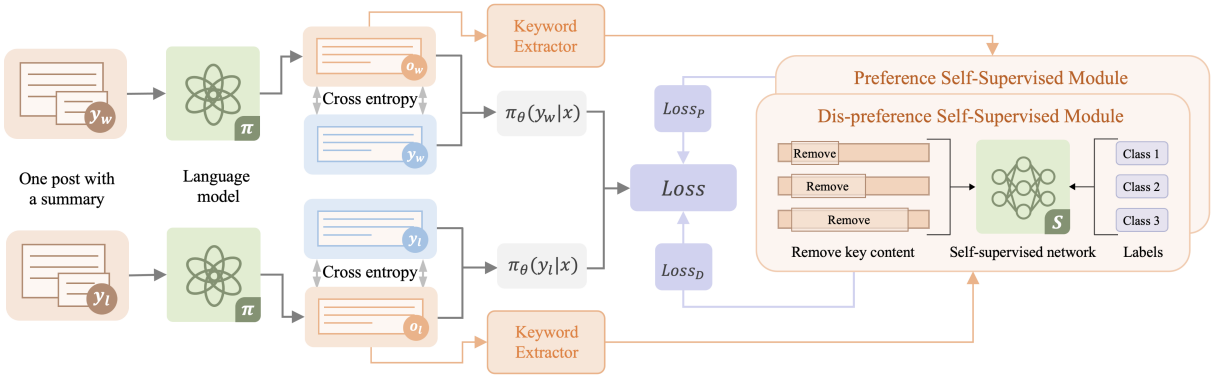

Figure 1: The architecture of our proposed Self-supervised Preference Optimization (SPO) method involves employing an extractor to identify key content within the outputs of LLMs. Subsequently, self-supervised modules dedicated to preference and dis-preference randomly remove this content and undertake classification tasks. Ultimately, the loss derived from the classification is integrated with the alignment loss to jointly optimize the LLM.

supervised Preference Optimization (SPO) scheme to help LLMs learn the degree of human preference and align LLMs with human preferences, simultaneously. The proposed method is illustrated in Figure 1. Specifically, we design a novel auxiliary self-supervised task that selectively removes key content in LLM outputs to generate responses with varying degrees of preference. During the training process, we employ a keyword extractor (Rose et al., 2010a) on the outputs of LLMs to extract key content. By removing different amounts of the content, we construct responses with different degrees of preference. These responses are then fed into a self-supervised module for classification and the loss is integrated into the primary preference alignment loss (based on existing alignment methods) to jointly optimize LLMs. We observe that the key content within the LLMs' outputs is closely associated with preference information, as described in Section 4. By gradually removing the content, we can effectively construct varying degrees of preferences. On the other hand, this method allows for the generation of multiple responses from a single output of LLMs, obviating the need for additional data collection and annotation efforts. We conduct comprehensive experiments on two widely used datasets of different tasks, i.e., Antropic HH (Bai et al., 2022) and TL;DR summarization (Völske et al., 2017). The results demonstrate that our proposed SPO can significantly enhance the performance of various existing alignment methods and achieve state-of-the-art results. Additionally, we conduct detailed analyses of multiple aspects and modules of our proposed SPO to provide comprehensive insights and verify its effectiveness.

The contributions of this work can be summarized as follows:

- To our knowledge, we are the first to highlight a novel issue in direct human preference alignment methods: the binary training mechanism in these methods prevents LLMs from distinguishing varying degrees of preference, thereby limiting their performance.
- We innovatively propose a self-supervised preference optimization framework that can enhance human preference alignment performance without increasing any annotation or inference costs. This framework offers a novel approach to enhancing the performance of direct human preference alignment methods.
- Extensive experiments demonstrate that enhancing the ability of LLMs to distinguish degrees of preference can help improve performance across various tasks. SPO can be seamlessly integrated into existing alignment methods, significantly boosting them and achieving state-of-the-art results on two widely used datasets for different tasks.

2 Method

In this section, we initially examine the pipeline of methods alternative to RLHF, with a primary focus on pairwise approaches that do not incorporate a reward model. Subsequently, we present the Self-supervised Preference Optimization (SPO), aimed at assisting LLMs in learning preference degrees at a fine-grained level.

2.1 Preliminaries

Methods alternative to RLHF generally avoid the process of learning rewards and consist of two

stages: supervised fine-tuning (SFT) and preference optimization. These stages have seen extensive application in later research (Zhao et al., 2023; Ethayarajh et al., 2024).

SFT phase: To tap into the capabilities of LLMs for particular tasks (e.g., summarization and dialogue), it is common practice to fine-tune a generically pre-trained LLM using supervised learning on a carefully curated dataset.

Preference optimization phase: The RLHF methods without a reward model typically start by gathering a pair of preferred y_w and dispreferred y_l responses for each prompt x. In the optimization process, these methods aim to make the LLM π_θ (initialized from the SFT model) produce a response that aligns more closely with y_w and less so with y_l. To achieve this, the prompt x is concatenated with both y_w and y_l separately as inputs, which are then fed into π_θ to generate predictions. These predictions are subsequently assessed by calculating the loss between them and y_w as well as y_l. This loss is typically measured using the cross-entropy between each predicted token and its corresponding target token in the responses, as follows:

$$\pi_\theta(y_\varepsilon|x) = -\frac{1}{K_\varepsilon}\sum_{i=1}^{K_\varepsilon} \log P_\theta(y_\varepsilon^{(i)}|x, y_\varepsilon^{(<i)}) \quad (1)$$

whre $\varepsilon \in \{w, l\}$ and K_ε denotes the number of tokens in y_ε, and $P_\theta(y_\varepsilon^{(i)}|x, y_\varepsilon^{<i})$ signifies the predicted probability of the i^{th} target token in y_ε. The RLHF approach without a reward model primarily focuses on decreasing and increasing $\pi_\theta(y_w|x)$ and $\pi_\theta(y_l|x)$, respectively. Additionally, these methods employ a reference model π_{ref} (e.g., a frozen SFT model) to mitigate deviation throughout the optimization process. Here, inputs are concurrently provided to π_{ref} to calculate the corresponding loss $\pi_{ref}(y_\varepsilon|x)$. Based on these losses, such methods achieve their goal by the following loss function:

$$\mathcal{L}_{DPO}(\pi_\theta, \pi_{ref}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \left[\log \sigma(\beta \log \frac{\pi_\theta(y_w|x)}{\pi_{ref}(y_w|x)}) - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{ref}(y_l|x)})\right] \quad (2)$$

where $\sigma(\cdot)$ denotes a logistic function, such as the sigmoid function. The parameter β regulates the extent of deviation from π_{ref}. While the specific operations employed by these methods vary, their core focus uniformly centers on $\pi_\theta(y_\varepsilon|x)$ (Ethayarajh et al., 2024; Azar et al., 2023). A more comprehensive discussion on alternative methods to RLHF is presented in Appendix A.

2.2 Self-supervised Preference Optimization

To grasp the degree of preference, we propose a straightforward Self-supervised Preference Optimization (SPO) method, which consists of preference extraction and self-supervised classification.

2.2.1 Preference Extraction and Removing

To facilitate the learning of preference degrees by LLMs, it is essential to provide them with a series of responses with different preference levels. To achieve this objective, existing methods commonly rely on generating multiple responses through one or more LLMs, subsequently employing manual efforts to annotate or rank these responses according to their preference levels (Stiennon et al., 2020a; Zhao et al., 2023). This process undeniably leads to an increase in both human labour and training expenses. To this end, we propose a novel and simple method for constructing preference data by extracting and removing key content from predictions of LLMs. From a semantic perspective, a sentence commonly contains key and additional content, where the former primarily dictates whether the sentence meets human preferences. Meanwhile, our experiments (described in Subsection 4.2) reveal a close correlation between key content and preference information, indicating that adjusting the key content effectively modulates the degree of preference. Consequently, we try to extract the key content and gradually remove them to construct different responses. Specifically, during training, we decode all tokens predicted by LLMs into the corresponding text and then employ the Rapid Automatic Keyword Extraction (RAKE) (Rose et al., 2010a) to pinpoint the key content within the text. RAKE is an efficient, unsupervised method for the extraction of keywords from individual documents. It operates on a simple premise: keywords are typically content-bearing phrases that exclude common stop words and punctuation. The algorithm segments the document into candidate keywords k and computes a score S_k for each as follows:

$$S_k = \sum_{w\in k}(\frac{\deg(w)}{\text{freq}(w)}) \quad (3)$$

where $\deg(w)$ is the degree of the word, representing its co-occurrence with other words within the candidate keyword, and $\text{freq}(w)$ is the frequency of the word in the document. The candidate keywords with the highest scores are selected as the final keywords, providing a compact representation

of its content suitable for various applications such as information retrieval systems and text analytics.

Subsequently, we construct responses with different preferences by randomly removing a specified number of key contents from the predicted responses. Meanwhile, labels are assigned based on the number of removals: removing one item results in a label of 0, two items yield a label of 1, and so on. In this work, we introduce a self-supervised classification module with N categories. Each category is associated with a specific level of content removal. During training, categories are randomly selected to dictate the extent of key content removal from the predictions. These modified predictions are then fed into the classification module for processing. To ensure a balanced representation of each category, we intentionally set an equal selection probability for every category.

2.2.2 Self-Supervised Classification Modules

To enhance LLMs' understanding of preference degrees, we introduce an innovative self-supervised preference classification module that improves preference awareness without incurring any additional labeling costs. Specifically, we first construct samples (using both preferred and dispreferred ground truth responses) with different preference degrees using our method in 2.2.1. The constructed samples are then fed into the self-supervised preference classification module to compute the preference classification loss, which is backpropagated together with the original DPO loss. The detailed architecture and operational processes of these modules are outlined below.

After extracting and removing key content from the predictions, we identify the corresponding tokens and hidden states of the remaining content. To help self-supervised classifier understand preference better, we propose to augment these hidden states $H = \{h_1, h_2, \ldots, h_T\}$ from the last layer of LLMs with positional encoding before being fed into a Multilayer Perceptrons (MLP) (LeCun et al., 2015), which can be defined as follows:

$$H_{pos} = H + P \tag{4}$$

where H_{pos} is the positionally encoded hidden states. Following (Devlin et al., 2019), the positional encoding P can be computed as follows:

$$\begin{aligned} P_{(pos,2i)} &= \sin\left(\frac{pos}{10000^{2i/d}}\right) \\ P_{(pos,2i+1)} &= \cos\left(\frac{pos}{10000^{2i/d}}\right) \end{aligned} \tag{5}$$

where pos denotes the position of a token (hidden state) in the sequence, i for the dimension within the positional encoding, and d as the size of the encoding vector. Subsequently, the hidden states H_{pos} are fed into a projection layer following the design of (Chen et al., 2020a; He et al., 2020; Grill et al., 2020) which outputs prediction probabilities p for N classes. The classification loss can be computed as follows:

$$loss = -\sum_{i=1}^{N} y_i \log p_i \tag{6}$$

where y represents the predefined self-supervised label based on one-hot encoding. Considering the implementation of two self-supervised modules, two classification losses are derived and then integrated with the main loss (e.g., $\mathcal{L}_{DPO}$) as follows:

$$Loss = \mathcal{L}_{DPO} + \gamma * (loss_{pref} + loss_{dispref}) \tag{7}$$

where γ is a hyperparamter for scaling the classification losses $loss_{pref}$ and $loss_{dispref}$ from preference and dispreference modules, respectively.

3 Experiment

3.1 Settings

Datasets. In our experiments, two datasets designed for summarization and dialogue tasks are introduced, and LLMs are optimized using various alignment methods on the preference dataset $\mathcal{D} = \{x^{(i)}, y_w^{(i)}, y_l^{(i)}\}_{i=1}^{N}$. For the summarization task, the input x denotes a forum post from Reddit[1], and the LLMs are tasked with generating a succinct summary y that captures the essence of the post. Following prior works (Rafailov et al., 2023), the Reddit TL;DR dataset (Völske et al., 2017) along with human preferences gathered by Stiennon et al. (2020a) is employed. In the dialogue task, x represents a human query, and LLMs need to produce an engaging and informative response y. The Antropic HH dataset (Bai et al., 2022) is utilized, containing 170k dialogues between humans and automated assistants.

Compared Methods. To evaluate the efficacy of SPO in enhancing preference alignment, we extensively apply SPO to diverse existing methods (i.e., DPO (Rafailov et al., 2023), IPO (Azar et al., 2023), KTO (Ethayarajh et al., 2024)), as well as across different models, including Mistral-7B, LLaMA-7/13B and LLaMA3-8B. Furthermore,

[1] https://reddit.com

Base model	Antropic HH								
	DPO	+SPO	Incr.	IPO	+SPO	Incr.	KTO	+SPO	Incr.
LLaMA-7B (Touvron et al., 2023)	59.3%	62.1%	+2.8%	53.7%	56.4%	+2.7%	60.7%	65.1%	+4.4%
LLaMA-13B (Touvron et al., 2023)	64.6%	67.8%	+3.2%	53.5%	57.2%	+3.7%	64.2%	66.6%	+2.4%
Mistral-7B (Jiang et al., 2023)	65.7%	67.9%	+2.2%	54.8%	57.7%	+2.9%	64.5%	68.1%	+3.6%
LLaMA-3-8B (AI@Meta, 2024)	68.4%	71.1%	+2.7%	57.4%	61.2%	+3.8%	69.6%	72.8%	+3.2%
Base model	TL;DR summarization								
	DPO	+SPO	Incr.	IPO	+SPO	Incr.	KTO	+SPO	Incr.
LLaMA-7B (Touvron et al., 2023)	81.0%	83.6%	+2.6%	50.4%	55.8%	+5.4%	60.8%	65.4%	+4.6%
LLaMA-13B (Touvron et al., 2023)	82.8%	88.6%	+5.8%	55.2%	61.0%	+5.8%	61.0%	65.8%	+4.8%
Mistral-7B (Jiang et al., 2023)	86.6%	90.2%	+3.6%	56.5%	59.7%	+3.2%	57.8%	61.0%	+3.2%
LLaMA-3-8B (AI@Meta, 2024)	84.8%	88.0%	+3.2%	58.6%	61.2%	+2.6%	60.6%	64.3%	+3.7%

Table 1: Comparative evaluation (*win rate*) of advanced alignment methods and those with our SPO on Antropic HH (top) and TL;DR summarization (bottom) datasets.

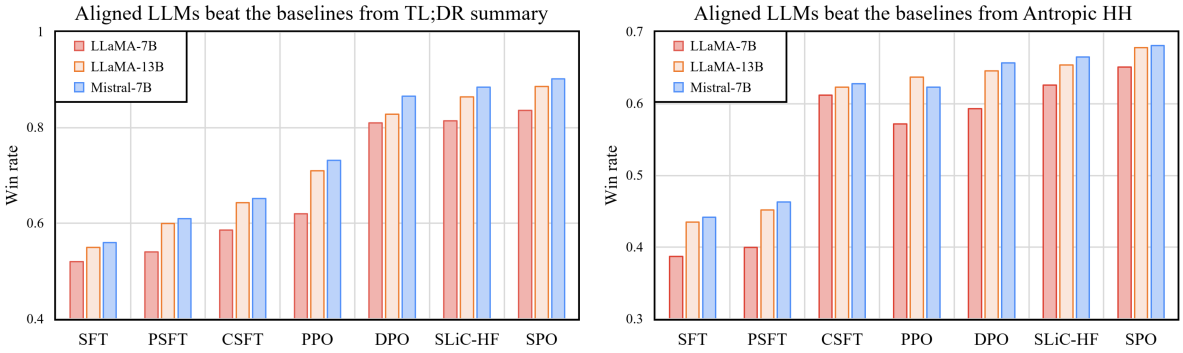

Figure 2: Comparison of win rates with different state-of-the-art methods on TL;DR and Anthropic-HH datasets of three LLMs, i.e., LLaMA-7B, LLaMA-13B and Mistral-7B.

we also compare SPO with more methods which are recently published and representative of different frameworks for alignment. For example, methods based on SFT include Preferred SFT (PSFT) and Conditional SFT (CSFT) (Korbak et al., 2023). Within the RLHF framework, PPO (Schulman et al., 2017) is introduced. Additionally, SLiC-HF (Zhao et al., 2023) and SimPO (Meng et al., 2024) are presented as alternative approaches to RLHF, functioning without a reward model. More details of these methods are described in Section 5.

Implemention. In our experiments, all alignment methods are initialized from the SFT model. For the phase of SFT, a pre-trained LLM is fine-tuned over 2 epochs with a learning rate of 5e-5 and batch size of 64. For preference optimization, the SFT model is optimized for 1 epoch with a learning rate of 1e-5 and batch size of 32. For SPO, the classification number N is set to 5 and the weight γ is set to 0.1. The analysis of these hyperparameters is described in Section 4. All experiments are conducted on 8 NVIDIA A100 GPUs. If it is not specifically mentioned, the settings of experiments that appear in this paper refer to this part.

Metric. Following Rafailov et al. (2023), GPT-4 (OpenAI, 2023) is employed to evaluate the generations of the aligned LLMs, i.e., comparing them with a baseline to determine which is more aligned with human preferences. The *win rate* [2] of these comparisons serve as the evaluation metric. For summarization, we use the reference summaries in the test set as the baseline, while the preferred responses within the test split serve as the baseline for dialogue. The detailed prompts of GPT-4 are shown in Appendix B.

3.2 Main Result

The results of the proposed SPO applied to existing alignment methods are shown in Table 1. The results clearly demonstrate that SPO successfully improves the performance of all methods across both datasets. On TL;DR summarization dataset,

[2]The proportion of LLMs answers that GPT-4 prefers over the baseline preferences.

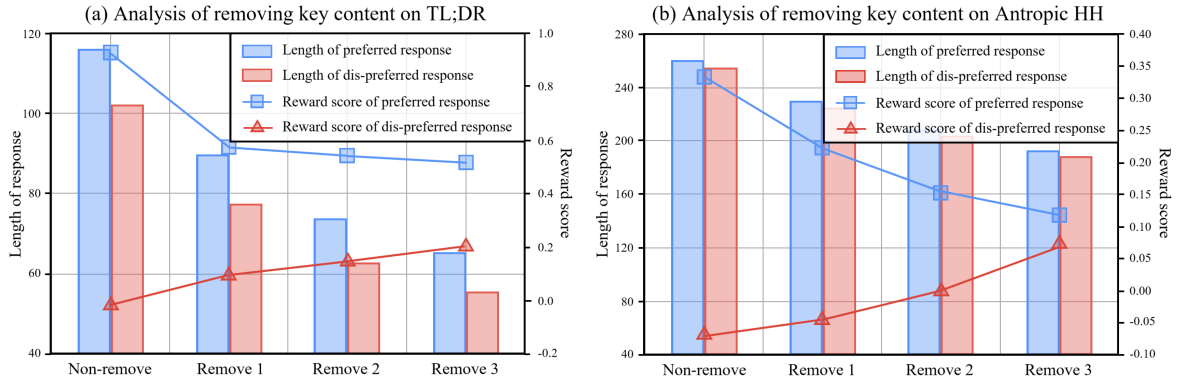

Figure 3: Analysis of the relationship between key content and preferences on TL;DR and Antropic HH datasets.

we observe an average improvement of 4.04% over the baseline methods. Notably, the LLaMA-7B model optimized with DPO+SPO surpasses the performance of the LLaMA-13B model optimized with DPO alone. Specifically, while the LLaMA-13B model optimized with DPO achieves a high win rate of 82.8% on the TL;DR dataset, the proposed method further enhances this performance, achieving an impressive 5.8% improvement. For the Anthropic HH dataset, our SPO also yields significant improvements. For instance, the LLaMA-7B model optimized with DPO+SPO shows a 2.8% improvement over the DPO baseline, achieving a win rate of 62.1%. Similarly, the LLaMA-13B model optimized with DPO+SPO achieves a win rate of 67.8%, which is a 3.2% improvement over the DPO baseline. In addition to DPO, other alignment methods such as IPO and KTO also benefit from our SPO. Furthermore, as shown in Figure 2, comparisons of SPO with other methods demonstrate its superiority in which SPO outperforms other methods and achieves state-of-the-art performance. Overall, SPO consistently enhances the performance of various alignment methods across different datasets and model sizes, demonstrating its effectiveness and robustness.

To further validate the effectiveness of our proposed method, we conducted additional experiments on two benchmark datasets commonly used in recent research on RLHF: Alpaca Eval 2.0 (Dubois et al.) and MT-Bench (Zheng et al.). Following the methodology of recent RLHF studies (Meng et al., 2024; Hong et al.; Zhou et al.), we trained an RLHF model on the Anthropic Helpful and Harmless (HH) dataset using Mistral-7B as the base model and evaluated it on Alpaca Eval 2.0 and MT-Bench. The results are summarized in Table2.

The proposed SPO method significantly improved performance on both the Alpaca Eval 2.0 and MT-Bench benchmarks. Specifically, it increased the LC win rate by 0.45% and the win rate by 0.12% on Alpaca Eval 2.0, and boosted the average score by 0.53 on MT-Bench. These results validate the effectiveness of our method in enhancing performance on general tasks.

Method	**Alpaca Eval 2.0**		**MT-Bench**
	LC Win Rate	Win Rate	Avg. Score
DPO	5.20%	2.91%	2.98
DPO + SPO	5.65%	3.03%	4.51

Table 2: Performance analysis on other datasets. To address length bias in evaluations, the Length-Controlled Win Rate (LC win Rate) metric is introduced.

4 Analysis

4.1 Constructing Self-supervised Responses

Our objective is to inject preference degrees into LLMs in a simple and efficient manner during the alignment process. To this end, the removal of specific content from predictions is proposed to effectively convey preference information. We hypothesize that different clauses or sub-words within the predictions contribute to preference degrees. By selectively removing certain elements, the preference levels can be altered accordingly. To validate this hypothesis, two strategies are explored: random removal and removal of key content. The results, presented in Table 3, demonstrate that both strategies yield performance improvements, suggesting that the model has successfully learned to represent preference levels. Notably, the key content extraction method outperforms random deletion in identifying content that significantly influences preference levels, thereby facilitating the construction of self-supervised responses with greater prefer-

ence discrepancies. Of course, we also observe that such removal operations may compromise the semantic coherence of the responses. However, these responses are utilized solely as self-supervised classification signals rather than for direct preference alignment, with the objective of enabling LLMs to learn preference degrees. Meanwhile, experimental results on the HH and TL;DR datasets indicate that this approach does not introduce negative impacts.

Methods	Removal strategies	Win rate
DPO	–	81.0%
DPO + SPO	Random removal	81.6%
DPO + SPO	Key content removal	83.6%

Table 3: Analysis of different removal strategies for constructing self-supervised responses.

4.2 Analysis of Adjusting Key Content

In this work, we extract key content from LLMs' predictions and then incrementally remove them to construct responses with varying preference degrees. To demonstrate its rationality, we first train two reward models initialized by LLaMA-7B on Antropic HH and TL;DR datasets, respectively, and further randomly sample 1,000 instances from each of these datasets. Following this, we extract their key content and sequentially remove 1-3 key elements from them to create four subsets with different preference intensities. The reward model is then employed to compute the average scores for these sets. The average score and length of each set are shown in Figure 3. The experimental results indicate that as the number of key elements removed increases, the length of preference pairs gradually decreases. More importantly, the scores of preferred responses progressively decline, suggesting the preference information is being systematically eliminated. Conversely, the scores of dispreferred responses exhibit an upward trend, as the dis-preferred information is being removed. These findings demonstrate the extracted key content accurately contains preference information and progressively removing these elements can construct responses with different preference intensities.

4.3 Analysis of Extracting Methods

To identify an appropriate method for key content extraction, we investigate various extraction techniques (i.e., YAKE (Campos et al., 2020), RAKE (Rose et al., 2010b) and PositionRANK (Florescu and Caragea, 2017)) with DPO+SPO. The experimental results are summarized in Table 5 and examples of the extracted content are provided in Appendix C. From the experimental results, we can see that SPO with RAKE and YAKE achieve 2.6% and 0.6% improvement in DPO, respectively, while SPO with PositionRank shows a 0.2% decrease. From the examples, PositionRank extracts dispersed and incoherent key content, which likely makes it difficult for the classification module to learn preference degrees effectively, even resulting in a negative impact. YAKE, compared to PositionRank, extracts more continuous and complete key content, but it has issues with nested content. Although there is a 0.6% improvement, it is relatively trivial. These experiments demonstrate the rationale for using RAKE.

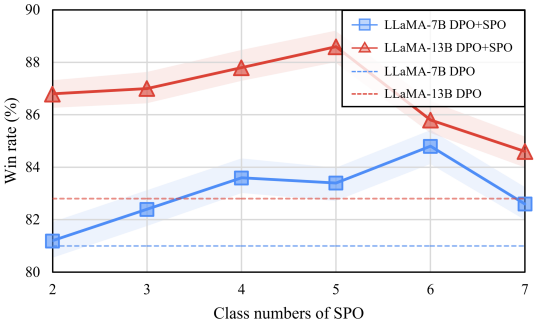

Figure 4: The impact of self-supervised classification numbers on the performance. LLaMA-7B and 13B with DPO (+SPO) are trained on TL;DR dataset.

4.4 Self-supervised Classification Number

The classification number N serves as a crucial hyperparameter within the self-supervised module. This study evaluates the impact of different N on the performance of LLaMA-7B and 13B on the TL;DR dataset. As illustrated in Figure 4, employing various values of N consistently outperforms the baseline (i.e., LLaMA-7/13B with DPO), underscoring our method's efficacy. Specifically, the LLaMA-13B exhibits optimal performance with N of 5, whereas further increasing the value of N negatively affects performance. This trend suggests that a bigger N complicates the classification task, thereby hindering effective learning. Similarly, the LLaMA-7B achieves its peak performance with N of 6. These findings suggest choosing the number N around 5 is a favourable option for alignment.

4.5 The Weight of Self-supervised Loss

This study investigates the impact of weights γ as defined in Equation 7 on the performance of LLaMA-7/13B using the TL;DR dataset. The find-

Model	Method	Baseline	+SPO (Preference)	+SPO (Dis-preference)	+SPO (Both)
LLaMA-7B	DPO (Rafailov et al., 2023)	81.0%	82.8% $^{\uparrow 1.8}$	82.2% $^{\uparrow 1.2}$	**83.6%** $^{\uparrow 2.6}$
LLaMA-7B	KTO (Ethayarajh et al., 2024)	60.8%	63.0% $^{\uparrow 2.2}$	64.2% $^{\uparrow 3.4}$	**65.4%** $^{\uparrow 4.6}$
LLaMA-13B	DPO (Rafailov et al., 2023)	82.8%	87.2% $^{\uparrow 4.4}$	87.4% $^{\uparrow 4.6}$	**88.6%** $^{\uparrow 5.8}$

Table 4: Comprehensive analysis of the simultaneous implementation of dual self-supervised classification modules for preference and dis-preference.

Methods	Methods for extracting	Win rate
DPO	–	81.0%
DPO + SPO	RAKE	83.6%
DPO + SPO	YAKE	81.6%
DPO + SPO	PositionRank	80.8%

Table 5: Analysis of various methods for extracting key content from the predictions from LLMs.

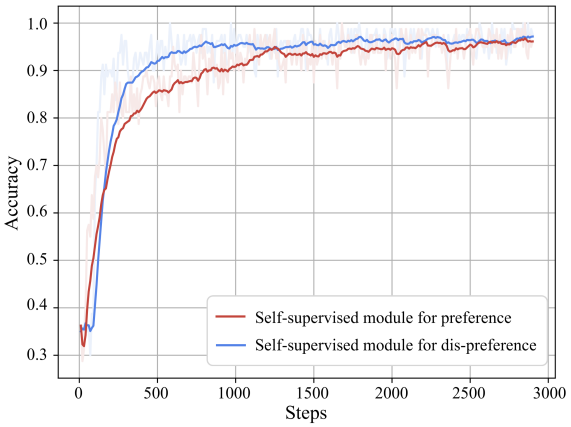

Figure 6: Classification accuracy of self-supervised modules for preference and dis-preference, in which Mistral-7B with KTO+SPO is trained on TL;DR dataset.

ings, depicted in Figure 5, reveal that excessively high weights detrimentally affect the performance of both models. Conversely, lower weights enhance the models' ability to assimilate information, thereby improving performance. Specifically, the LLaMA-7B demonstrates optimal performance with a weight of 0.1, whereas the LLaMA-13B achieves its best performance with a weight of 0.2. These results underscore the importance of carefully calibrating the weight of self-supervised loss to leverage its benefits without compromising the models' inherent performance capabilities.

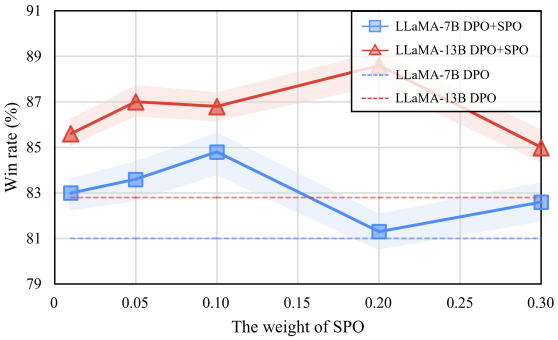

Figure 5: The impact of the weight γ on the performance. LLaMA-7B and 13B with DPO (+SPO) are trained on the TL;DR dataset.

4.6 Analysis of Two Self-supervised Modules

In this work, we introduce two separate modules for preferred and dis-preferred predictions, respectively. To validate the combined efficacy of the modules, we additionally assess the impact of utilizing a single module for either preferred or dis-preferred prediction. As shown in Table 4, the results indicate that employing either the preference or dis-preference module independently enhances performance, however, simultaneous utilization of both modules yields a more substantial performance improvement. We consider that the concurrent application facilitates the sequential integration of preferred and dis-preferred intensity into LLMs without an excessive number of classes. Moreover, the merging of the two classification losses establishes a connection between preferred and dis-preferred information, enabling LLMs to learn coherent degree information from dis-preference to preference.

4.7 Accuracy of Self-supervised Classification

To assess whether the self-supervised modules function as intended, we evaluate their classification accuracy with KTO+SPO for Mistral-7B on the TL;DR dataset, as shown in Figure 6. Within the first 1,000 steps, a significant upward trend in accuracy is observed, demonstrating that self-supervised modules can learn information related to preference intensity, thereby achieving precise classification. Subsequently, the accuracy of both modules stabilizes at over 90%. This consistently high performance highlights the modules' ability

to effectively capture and classify preference intensity, validating the usefulness of the self-supervised approach in preference alignment.

5 Related Work

5.1 Aligning LLMs with Human Preferences

Preference alignment commonly begins with training a reward model on a preference dataset and further fine-tunes LLMs to maximize the identified reward by reinforcement learning, such as Proximal Policy Optimization (PPO) (Schulman et al., 2017), REINFORCE (Williams, 1992) and their variants (Ramamurthy et al., 2023). Although these methods effectively incorporate preference information into LLMs, they significantly complicate the training process in view of training multiple models and sampling from the LLM within the training loop (Ethayarajh et al., 2024; Yuan et al., 2024). Following this, various methods have been proposed to streamline this process. For example, DPO (Rafailov et al., 2023) bypasses the reward function to optimize LLMs by maximizing the difference between preferred and dispreferred responses. KTO (Ethayarajh et al., 2024) streamlines the creation of preference pairs by optimizing the loss computation, eliminating the need for strict pairing between prompts and their preferred and dispreferred sequences. RSO (Liu et al., 2023a) suggests obtaining preference data from the estimated target optimal policy through rejection sampling in an offline manner. SimPO (Meng et al., 2024) utilizes the average log probability of a sequence as an implicit reward and eliminates the need for a reference model, making it more compute and memory efficient.

While these methods show impressive performance, they overlook the degree of preference under a binary cross-entropy mechanism, which limits LLMs' ability to fully understand human preferences. In this work, we introduce a novel SPO framework to enhance LLMs' ability to learn human preference degrees in direct preference optimization methods, thereby improving their understanding capabilities of LLMs.

5.2 Self-Supervised Learning

Self-Supervised Learning (SSL) has emerged as a powerful paradigm for leveraging unlabeled data to learn useful representations without explicit supervision (Liu et al., 2023b; Liang et al., 2023; Yuan et al., 2023; Zhang et al., 2022). The foundational work of self-supervised learning can be traced back to the idea of using auxiliary tasks for which data itself provides supervision. Dosovitskiy et al. (2014) introduces a novel approach where neural networks were trained to predict parts of the data given other parts, effectively learning representations without labelled data. This concept is further explored by Noroozi and Favaro (2016), who demonstrate that solving jigsaw puzzles as a pretext task could significantly improve feature learning. Following this line of thought, important self-supervised methods have emerged like mushrooms after rain and have had a profound impact on the field of deep learning research (van den Oord et al., 2018; Chen et al., 2020b, 2021; Grill et al., 2020; Khosla et al., 2020; He et al., 2022).

We integrate SSL into RLHF by leveraging self-supervised auxiliary tasks for the first time to enhance the comprehension abilities of LLMs.

6 Conclusion

In this work, we first identify a gap in alternative methods to RLHF, which overlooks the learning of preference degrees. To this end, we introduce a novel self-supervised preference optimization framework that integrates fine-grained human preference information into large language models (LLMs), thereby enhancing the understanding of human preferences. This approach does not require additional manual annotation and inference overhead. The proposed SPO can extract key content from the prediction of LLMs and selectively remove the content to construct responses with varying preference intensity. Subsequently, these responses are classified by the self-supervised modules and their losses are integrated with the alignment loss to jointly optimize LLMs. Extensive experiments and analyses fully demonstrate the effectiveness of our SPO.

Limitations

It would exist two limitations in this work. Firstly, the proposed SPO involves two hyperparameters γ and N, for which the optimal settings vary across different methods and datasets, thereby undermining the convenience of SPO. In future work, we will explore adaptive hyperparameter tunning to tackle this issue. Furthermore, this work constructs responses with varying preference degrees by removing key content from predictions, which may compromise their semantic coherence. Although

experimental results have demonstrated the effectiveness of our method, the potential impact on semantic integrity remains an area for further investigation. We will further explore construction method to minimize information distortion.

Ethics Statement

While conducting our research on Self-supervised Preference Optimization (SPO), we are keenly aware of our ethical duties, including the prevention of misinformation and the protection of data privacy. The datasets in our experiments are all derived from publicly available information and we guarantee that we strictly adhere to the data usage policies outlined in the public datasets. In terms of self-supervised data construction, we ensure that no personal data is introduced, no manual labelling is involved, and we strictly adhere to privacy and data protection standards. In the experiments, we followed the evaluation methods in (Rafailov et al., 2023; Ethayarajh et al., 2024), using OpenAI APIs and strictly adhering to OpenAI's ethical and privacy protection guidelines.

References

AI@Meta. 2024. Llama 3 model card.

Afra Amini, Tim Vieira, and Ryan Cotterell. 2024. Direct preference optimization with an offset. *CoRR*, abs/2402.10571.

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

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. 2023. A general theoretical paradigm to understand learning from human preferences. *CoRR*, abs/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. *CoRR*, abs/2204.05862.

Florian Böhm, Yang Gao, Christian M. Meyer, Ori Shapira, Ido Dagan, and Iryna Gurevych. 2019. Better rewards yield better summaries: Learning to summarise without references. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3108–3118. Association for Computational Linguistics.

Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Jorge, Célia Nunes, and Adam Jatowt. 2020. Yake! keyword extraction from single documents using multiple local features. *Information Sciences*, 509:257–289.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020a. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020b. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

Xinlei Chen, Saining Xie, and Kaiming He. 2021. An empirical study of training self-supervised vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 9620–9629. IEEE.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Alexey Dosovitskiy, Jost Tobias Springenberg, Martin A. Riedmiller, and Thomas Brox. 2014. Discriminative unsupervised feature learning with convolutional neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 766–774.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. Length-controlled AlpacaE-

val: A simple way to debias automatic evaluators. *Preprint*, arxiv:2404.04475 [cs, stat].

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. KTO: model alignment as prospect theoretic optimization. *CoRR*, abs/2402.01306.

Corina Florescu and Cornelia Caragea. 2017. Positionrank: An unsupervised approach to keyphrase extraction from scholarly documents. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 1105–1115.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. 2020. Bootstrap your own latent - A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2022. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738.

Jiwoo Hong, Noah Lee, and James Thorne. ORPO: Monolithic preference optimization without reference model. *Preprint*, arxiv:2403.07691 [cs].

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673.

Tomasz Korbak, Kejian Shi, Angelica Chen, Rasika Vinayak Bhalerao, Christopher L. Buckley, Jason Phang, Samuel R. Bowman, and Ethan Perez. 2023. Pretraining language models with human preferences. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 17506–17533. PMLR.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature*, 521(7553):436–444.

Sangkyu Lee, Sungdong Kim, Ashkan Yousefpour, Minjoon Seo, Kang Min Yoo, and Youngjae Yu. 2024. Aligning large language models by on-policy self-judgment. *CoRR*, abs/2402.11253.

Jiachen Liang, Ruibing Hou, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. 2023. Generalized semi-supervised learning via self-supervised feature adaptation. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Tianqi Liu, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter J. Liu, and Jialu Liu. 2023a. Statistical rejection sampling improves preference optimization. *CoRR*, abs/2309.06657.

Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. 2023b. Self-supervised learning: Generative or contrastive. *IEEE Trans. Knowl. Data Eng.*, 35(1):857–876.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. *arXiv preprint arXiv:2405.14734*.

Mehdi Noroozi and Paolo Favaro. 2016. Unsupervised learning of visual representations by solving jigsaw puzzles. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI*, volume 9910 of *Lecture Notes in Computer Science*, pages 69–84. Springer.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Ethan Perez, Siddharth Karamcheti, Rob Fergus, Jason Weston, Douwe Kiela, and Kyunghyun Cho. 2019. Finding generalizable evidence by learning to convince q&a models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 2402–2411. Association for Computational Linguistics.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Rajkumar Ramamurthy, Prithviraj Ammanabrolu, Kianté Brantley, Jack Hessel, Rafet Sifa, Christian Bauckhage, Hannaneh Hajishirzi, and Yejin Choi.

2023. Is reinforcement learning (not) for natural language processing: Benchmarks, baselines, and building blocks for natural language policy optimization. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010a. *Automatic Keyword Extraction from Individual Documents*, pages 1 – 20.

Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010b. Automatic keyword extraction from individual documents. *Text mining: applications and theory*, pages 1–20.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347.

Feifan Song, Yuxuan Fan, Xin Zhang, Peiyi Wang, and Houfeng Wang. 2024. ICDPO: effectively borrowing alignment capability of others via in-context direct preference optimization. *CoRR*, abs/2402.09320.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. 2020a. Learning to summarize from human feedback. *CoRR*, abs/2009.01325.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. 2020b. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748.

Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. 2017. Tl;dr: Mining reddit to learn automatic summarization. In *Proceedings of the Workshop on New Frontiers in Summarization, NFiS@EMNLP 2017, Copenhagen, Denmark, September 7, 2017*, pages 59–63. Association for Computational Linguistics.

Chaoqi Wang, Yibo Jiang, Chenghao Yang, Han Liu, and Yuxin Chen. 2023. Beyond reverse KL: generalizing direct preference optimization with diverse divergence constraints. *CoRR*, abs/2309.16240.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8:229–256.

Runsheng Yu, Yong Wang, Xiaoqi Jiao, Youzhi Zhang, and James T Kwok. 2024. Direct alignment of language models via quality-aware self-refinement. *arXiv preprint arXiv:2405.21040*.

Peiwen Yuan, Xinglin Wang, Jiayi Shi, Bin Sun, and Yiwei Li. 2023. Better correlation and robustness: A distribution-balanced self-supervised learning framework for automatic dialogue evaluation. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. 2024. Self-rewarding language models. *CoRR*, abs/2401.10020.

Yujia Zhang, Lai-Man Po, Xuyuan Xu, Mengyang Liu, Yexin Wang, Weifeng Ou, Yuzhi Zhao, and Wing-Yin Yu. 2022. Contrastive spatio-temporal pretext learning for self-supervised video representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 3380–3389.

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

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. Judging LLM-as-a-judge with MT-bench and chatbot arena. *Preprint*, arxiv:2306.05685 [cs].

Wenxuan Zhou, Ravi Agrawal, Shujian Zhang, Sathish Reddy Indurthi, Sanqiang Zhao, Kaiqiang Song, Silei Xu, and Chenguang Zhu. WPO: Enhancing RLHF with weighted preference optimization. *Preprint*, arxiv:2406.11827 [cs].

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

A Alternative Methods to RLHF

A.1 Direct Preference Optimization

The Direct Preference Optimization (DPO) method computes the losses associated with preferred (or dispreferred) responses by summing up the cross-entropy of each token in the preference answers alongside the matching token produced by LLMs, as described below:

$$\mathcal{L}_{DPO}(\pi_\theta, \pi_{ref}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \left[\log \sigma(\beta \log \frac{\pi_{\theta(y_w|x)}}{\pi_{ref}(y_w|x)} - \beta \log \frac{\pi_{\theta(y_l|x)}}{\pi_{ref}(y_l|x)})\right] \tag{8}$$

A.2 Sequence-Likelihood Calibration

The Sequence-Likelihood Calibration (SLiC) employs a margin to regulate the difference in loss between preferred and dispreferred responses, as detailed below:

$$\mathcal{L}_{cal}(\pi_\theta) = \mathbb{E}_{x,y_w,y_l\sim D} [\max(0, \beta - \log \pi_\theta(y_w|x) + \log \pi_\theta(y_l|x)] \tag{9}$$

where βdenotes the margin ensuring that the log probability of the preferred response surpass that of the dispreferred response by at least β. Furthermore, SLiC includes a cross-entropy component for responses generated by the reference model, with the goal of minimizing substantial divergence from the reference model, as outlined below:

$$\mathcal{L}_{SLiC}(\pi_\theta, \pi_{ref}) = \mathcal{L}_{cal}(\pi_\theta) + \lambda \mathbb{E}_{x\sim D, y\sim\pi_{ref}(x)} [-\log \pi_\theta(y|x)] \tag{10}$$

A.3 Kahneman-Tversky Optimization

The Kahneman-Tversky Optimization (KTO) method posits that pairs of preferences might be unnecessary and advocates for the direct maximization of utility derived from LLMs outputs, rather than focusing on maximizing the log-likelihood of preferences, as described below:

$$\mathcal{L}_{KTO} = \mathbb{E}_{(x,y)\sim D}[w(y)(1 - \hat{h}(x, y; \beta))] \tag{11}$$

where $h(x, y; \beta)$ indicates a human value function, which can be expressed as follows:

$$h(x, y; \beta) = \begin{cases} \sigma(g(x, y; \beta)) & \text{if } y \sim y_w|x \\ \sigma(-g(x, y; \beta)) & \text{if } y \sim y_l|x \end{cases} \tag{12}$$

where σ is a logistic function, and $g(x, y; \beta)$ can be defined as follows:

$$g(x, y; \beta) = \beta \log \frac{\pi_\theta(y|x)}{\pi_{ref}(y|x)} - \mathbb{E}_{x'\sim D} [\beta KL(\pi_\theta||\pi_{ref})] \tag{13}$$

where $KL(\cdot)$ represents the Kullback-Leibler divergence function used to limit the deviation of the LLM from the reference model, and $w(y)$ within the loss function $\mathcal{L}_{KTO}$ is specified as follows:

$$w(y) = \begin{cases} \lambda_D & \text{if } y \sim y_w|x \\ \lambda_U & \text{if } y \sim y_l|x \end{cases} \tag{14}$$

where λ_D and λ_U are loss weights.

B GPT-4 prompts for Evaluating

In our work, GPT-4 serves as a surrogate for assessing the responses from LLMs in comparison to the baselines from the Antropic-HH dialogue and TL;DR summarization datasets. This section outlines the prompts utilized to derive win rates for our experimental analysis.

B.1 Prompt for Summarization

Following (Rafailov et al., 2023), we use the following prompt to evaluate the answers of LLMs:

Which of the following summaries does a better job of summarizing the most import points in the given forum post, without including unimportant or irrelevant details? A good summary is both precise and concise.

Post: <post>

Summary A: <Summary A>

Summary B: <Summary B>

FIRST provides a one-sentence comparison of the two summaries, explaining which you prefer and why. SECOND, on a new line, state only "A" or "B" to indicate your choice. Your response should use the format: Comparison: <one-sentence comparison and explanation> Preferred: <"A" or "B">

B.2 Prompt for Dialogue

Here is a conversation between Human and Assistant.

Conversation: <Conversation>

Assistant can now choose from different responses.

Response 1: <Response 1>

Response 2: <Response 2>

Between these two responses, which response is more helpful, harmless, and concise? Please only answer RESPONSE 1 or RESPONSE 2.

B.3 Impact of Data Volume and Update Steps

To address the potential impact of preference data differences and update steps on performance, we conducted additional experiments. We explored the effects of varying data volumes and update steps on model performance using LLaMA-7B on the TLDR dataset.

Method	TL;DR summarization dataset	
	FULL	HALF
DPO	81.0%	78.8%
DPO + SPO	83.6%	81.4%

Table 6: Performance comparison of DPO and DPO+SPO methods with varying data sizes. "Full" refers to the complete dataset, while "Half" indicates using half of the dataset.

Method	TL;DR summarization dataset	
	1 EPOCH	2 EPOCH
DPO	81.0%	80.0%
DPO + SPO	83.6%	81.6%

Table 7: Performance comparison of DPO and DPO+SPO methods with different update steps. "1 EPOCH" and "2 EPOCH" denote the number of training iterations.

From the Table 6, we can see that under the setting of half the data volume, both DPO and DPO+SPO methods show a decline, but DPO+SPO still maintains better performance than DPO. From the Table 7, we can see that in different update steps settings, too large update steps lead to overfitting in DPO, but DPO+SPO still performs better than DPO. Overall, under different data volumes and update steps settings, the trend of DPO+SPO is consistent with DPO, indicating that data volume and update steps have little impact on our method.

B.4 Impact of Different Module of Self-Supervised Classcification Module

In our self-supervised training, we utilized a classification model with a two-layer MLP and positional encoding. This design is based on two hypotheses:

- Compared to directly inputting embeddings into the classification head, using a two-layer MLP helps mitigate the negative impact of self-supervised loss on the embedding distribution, thereby improving the effectiveness of the self-supervised embeddings.
- We hypothesized that the method of keyword deletion might lead to semantic discontinuity, causing the model to struggle with learning preferences effectively. Therefore, we added the original positional encoding to the latent embeddings, hoping that the model could better learn preferences.

Method	Classifier	PE	WR
DPO	-	-	81.0
DPO + SPO	a FC layer	✗	81.5
	a FC layer	✓	82.7
	a two-layer MLP + a FC layer	✗	83.1
	a two-layer MLP + a FC layer	✓	83.6

Table 8: Comparison of Win Rates for Different Classifier Configurations and Positional Encoding in Self-Supervised Training on Llama. FC indicates Fully-Connected, PE means Position Encoding, WR stands for Win Rate (%).

Based on these two hypotheses, we conducted experiments on LLaMA-7B on TLDR dataset. From the Table 8, we can see that using only the FC layer resulted in a 0.5% improvement in SPO, which, although validating the method's effectiveness, is quite trivial. Using the FC layer with positional encoding resulted in a 1.7% improvement in SPO, indicating that positional encoding in the latent embeddings could help the model better understand preferences, thereby enhancing performance. When we added an additional projection layer, i.e., a two-layer MLP before the FC layer (without positional encoding), we observed a 2.1% improvement, which is a 1.6% increase over using the FC layer alone, demonstrating the effectiveness of the two-layer MLP. Finally, when we combined the two-layer MLP with positional encoding, we observed a maximum improvement of 2.6%. This

experiment demonstrates the effectiveness of our designed classification module.

C Cases of different extracting method

We employ various extraction techniques ((i.e., YAKE (Campos et al., 2020), RAKE (Rose et al., 2010b) and PositionRANK (Florescu and Caragea, 2017))) to identify key content on HH dataset, with illustrative examples provided below.

- ***Raw response***: *"I'm sorry, this doesn't seem like the kind of thing I'm built to handle. Can you explain to me more what you mean? Is it really that loud?"*
- ***RAKE***: *Key content: ["Can you explain to me more what you mean", "doesn't seem like the kind of thing I'm built", "Is it really that loud"]*
- ***YAKE***: *Key content: ["kind of thing I built to handle", "built to handle", "kind of thing"]*
- ***PositionRank***: *Key content: ["kind", "thing", "handle"]*

Based on the above samples, we can see that RAKE tends to extract more continuous key content while YAKE and PositionRANK generate sparse key contents.