Ambiguity Awareness Optimization: Towards Semantic Disambiguation for Direct Preference Optimization

Jian Li[1,*], **Shenglin Yin**[2,*], **Yujia Zhang**[1,†,‡], **Alan Zhao**[1,†],
Xi Chen[1,†], **Xiaohui Zhou**[1], **Pengfei Xu**[1]

[1]AI Technology Center of OVB, Tencent, China
[2]School of Computer Science, Peking University, China

Abstract

Direct Preference Optimization (DPO) is a widely used reinforcement learning from human feedback (RLHF) method across various domains. Recent research has increasingly focused on the role of token importance in improving DPO effectiveness. It is observed that identical or semantically similar content (defined as ambiguous content) frequently appears within the preference pairs. We hypothesize that the presence of ambiguous content during DPO training may introduce ambiguity, thereby limiting further improvements in alignment. Through mathematical analysis and proof-of-concept experiments, we reveal that ambiguous content may potentially introduce ambiguities, thereby degrading performance. To address this issue, we introduce Ambiguity Awareness Optimization (AAO), a simple yet effective approach that automatically re-weights ambiguous content to reduce ambiguities by calculating semantic similarity from preference pairs. Through extensive experiments, we demonstrate that AAO consistently and significantly surpasses state-of-the-art approaches in performance, without markedly increasing response length, across multiple model scales and widely adopted benchmark datasets, including AlpacaEval 2, MT-Bench, and Arena-Hard. Specifically, AAO outperforms DPO by up to 8.9 points on AlpacaEval 2 and achieves an improvement of by up to 15.0 points on Arena-Hard.

1 Introduction

To align large language models (LLMs) with human values and intentions, learning from human feedback is essential. (Ziegler et al., 2019; Stiennon et al., 2020; Bai et al., 2022b). Reinforcement learning from human feedback (RLHF) has emerged as a key approach for aligning LLMs with human preferences and values (Christiano et al., 2017; Ouyang et al., 2022). The core of this technique lies in the introduction of an explicit reward model during training to generate reward signals, and the application of reinforcement learning under a reference model that is consistent with the initial policy (Achiam et al., 2023; Touvron et al., 2023).

While RLHF enhances output quality, its reliance on multiple models and iterative sampling increases training complexity (Zhao et al., 2023; Yuan et al., 2023; Azar et al., 2024; Hong et al., 2024; Casper et al., 2023). To address this, direct alignment approaches like Direct Preference Optimization (DPO) (Rafailov et al., 2023) and its variants (Zhao et al., 2023; Yuan et al., 2023; Hong et al., 2024; Ethayarajh et al., 2024b; Park et al., 2024; Xu et al., 2024; Ethayarajh et al., 2024a; Tang et al., 2024; Meng et al., 2024) directly optimize LLMs based on human preferences, bypassing the need for separate reward models. These methods adjust the model's loss by favoring preferred responses and penalizing dispreferred ones (Zhao et al., 2024; Zeng et al., 2024; Liao et al., 2024; Liu et al., 2024). However, DPO primarily emphasizes sequence-level preferences, overlooking the varying importance of individual tokens, which limits its effectiveness (Lin et al., 2024; Zeng et al., 2024; Liu et al., 2024; Gu et al., 2025).

To investigate why distinguishing the importance of tokens is critical in DPO, we observe that many training pairs contain identical or semantically similar words (which we define as ambiguous contents). We hypothesize that training on ambiguous contents may lead to confusion, thereby limiting the performance of DPO.

Building on this insight, we propose a simple yet effective method called Ambiguity Awareness Optimization (AAO) (shown in Figure 1), which can re-weight tokens to mitigate ambiguity during training by LLM itself. Specifically, we categorize

[*]Equal contribution.
[†]Corresponding author.
[‡]Project lead.
[§]Our code may be found at: https://github.com/InsLin/AAO.

response tokens into three types based on their semantic similarity (as measured by the LLM's own embeddings), sorted from highest to lowest similarity: ambiguous tokens, transitional tokens, and key tokens. For each token type, we design distinct weight adjustment curves according to their semantic similarity. Furthermore, we introduce an adaptive module in the latent space that automatically determines the decision thresholds for the three token groups, enabling the model to dynamically decide these thresholds during training.

Extensive experiments were conducted on Llama3.1-8B (Grattafiori et al., 2024) and Mistral-7B (Jiang et al., 2024) models, including both base and instruction-tuned variants, utilizing widely adopted benchmark datasets such as AlpacaEval 2 (Li et al., 2023a; Dubois et al., 2024), MT-bench (Zheng et al., 2023), and Arena-Hard (Li et al., 2024), among others. The results demonstrate that AAO consistently and significantly enhances the performance of DPO. Additionally, we further validated that AAO can alleviate the "squeeze effect" (Ren and Sutherland, 2024) present in DPO, revealing that the "squeeze effect" arises not only from sequence-level semantically similar pairs, but also from ambiguous tokens within the pairs. It is noteworthy that AAO does not rely on external models or additional data, making it highly flexible and easy to deploy.

Our work makes three key contributions:

- We conduct a thorough mathematical analysis of the effect of ambiguous content and perform comprehensive experiments, revealing the negative impact of ambiguous content on preference optimization.

- We propose a simple yet effective method called AAO, which can automatically identify ambiguous content during training. Additionally, it can be seamlessly integrated with existing methods, enhancing their performance without much computational burden.

- Extensive experiments demonstrate that applying AAO to existing methods results in significant performance improvements and achieves state-of-the-art results on four popular benchmarks of different tasks, i.e., AlpacaEval 2, Arena-Hard, MT-bench, Llama-Guard.

2 Method

2.1 Preliminaries

The RLHF method without a reward model starts by collecting preferred y_w and dispreferred y_l answers for each prompt x. These methods guide the LLM π_θ (initialized from the SFT model) to generate responses closer to y_w and farther from y_l. The prompt x is concatenated with y_w and y_l as inputs to π_θ, which outputs predictions. The loss is then computed as the product of the predicted probabilities for the target tokens, as follows:

$$\pi_\theta(y_\varepsilon|x) = \prod_{i=1}^{K_\varepsilon} P_\theta(y_\varepsilon^{(i)}|x, y_\varepsilon^{(<i)}), \tag{1}$$

where $\varepsilon \in \{w, l\}$, K_ε is the number of tokens in answer y_ε, and $P_\theta(y_\varepsilon^{(i)}|x, y_\varepsilon^{(<i)})$ denotes the predicted likelihood for the i^{th} target token in y_ε. The RLHF method without a reward model (e.g., DPO) aims to decrease $\pi_\theta(y_w|x)$ and increase $\pi_\theta(y_l|x)$. It also employs a reference model π_{ref} (e.g., a frozen SFT model) to mitigate alignment deviation, using simultaneous inputs to obtain the loss factor $\pi_{ref}(y_\varepsilon|x)$. Based on these factors, this approach achieves its objective with the following function:

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

where $\sigma(\cdot)$ denotes a logistic function (e.g., sigmoid). The parameter β controls the deviation from π_{ref}. Although the specific operations of these methods are different, they all ultimately revolve around $\pi_\theta(y_\varepsilon|x)$ (Ethayarajh et al., 2024a; Azar et al., 2024).

2.2 Theoretical Analysis of AAO

Through reviewing existing alternative methods to RLHF, we note that their implementations mainly rely on the preference loss factor $\pi_\theta(y_\varepsilon|x)$. The mechanism of logarithmic subtraction of these factors enables LLMs to generate preferred answers. As reported in (Rafailov et al., 2023), the gradient of the loss function $\mathcal{L}_{DPO}$ increases the likelihood of the preferred answers y_w and decreases that of the dispreferred answers y_l. Take a close look at Eq. 2, which can be reformulated as follows:

$$\mathcal{L}_{DPO}(\pi_\theta, \pi_{ref}) = -\mathbb{E}_{x,y_w,y_l\sim\mathcal{D}} \left[\log\sigma(\beta\log(\frac{\pi_{ref}(y_l|x)}{\pi_{ref}(y_w|x)} \cdot \frac{\pi_\theta(y_w|x)}{\pi_\theta(y_l|x)}))\right], \tag{3}$$

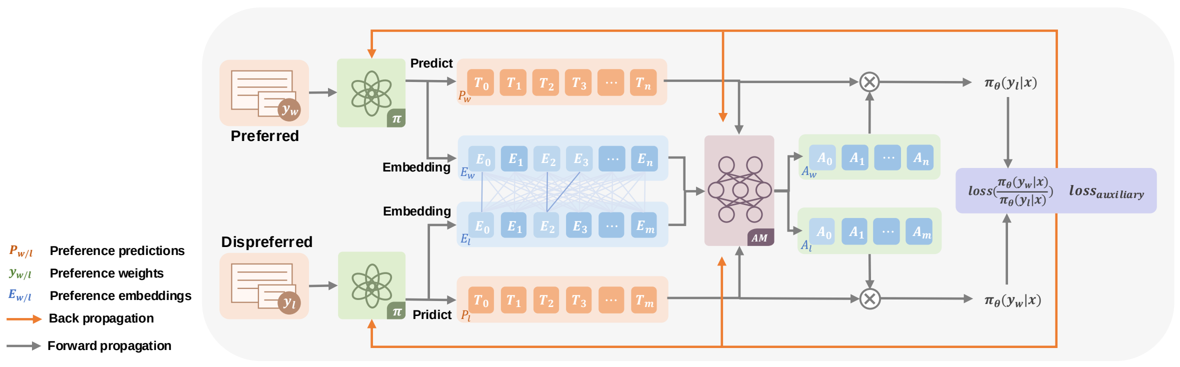

Figure 1: Diagram of our proposed AAO, in which background tokens among preference answers are re-weighted when computing cross-entropy loss. Firstly, AAO tokenizes preference pairs using a language model and encoding them into corresponding embeddings. Then AAO calculates the semantic similarity of the embeddings with cosine distance and decides the background tokens with a adaptive threshold. Finally, the background tokens are re-weighted during training.

where $\frac{\pi_{ref}(y_l|x)}{\pi_{ref}(y_w|x)}$ is a constant scaling factor due to the reference model without the need for gradient updates, while $\frac{\pi_\theta(y_w|x)}{\pi_\theta(y_l|x)}$ serves as the core component and enables LLMs to effectively achieve preference optimization. According to Eq. 1, the latter can be further transformed as follows:

$$\mathcal{L}_{core}(\pi_\theta, x, y_w, y_l) = \frac{\prod_{i=1}^{K_w} P_\theta(y_w^{(i)}|x, y_w^{(<i)})}{\prod_{i=1}^{K_l} P_\theta(y_l^{(i)}|x, y_l^{(<i)})}$$
$$= \sum_{i=1}^{K_w} \log P_\theta(y_w^{(i)}|x, y_w^{(<i)}) - \sum_{j=1}^{K_l} \log P_\theta(y_l^{(j)}|x, y_l^{(<j)}). \tag{4}$$

In the preferred answer y_w and the non-preferred answer y_l, if there exists a common token $y^{(c)}$ appearing at position i in the preferred answer and at position j in the non-preferred answer, the corresponding conditional probabilities are represented as $P_\theta(y^{(c)}|x, y_w^{(<i)})$ and $P_\theta(y^{(c)}|x, y_l^{(<j)})$, respectively. Under the optimization objective of Eq. 4, the contribution term of the common token $y^{(c)}$to the log-probability difference is expressed as follows:

$$\log P_\theta(y^{(c)}|x, y_w^{(<i)}) - \log P_\theta(y^{(c)}|x, y_l^{(<j)}) \tag{5}$$

Since $y^{(c)}$ appears in different contexts in the preferred path and the non-preferred path, specifically $y_w^{(<i)} \neq y_l^{(<j)}$, the model generates conflicting gradient signals for the prediction of the same token:

- In the preferred path, the optimization pushes $\log P_\theta(y^{(c)}|x, y_w^{(<i)})$ to increase in order to boost the overall probability of the preferred path.
- In the non-preferred path, the optimization drives $\log P_\theta(y^{(c)}|x, y_l^{(<j)})$ to decrease in order to reduce the probability of the non-preferred path.

However, since$y^{(c)}$ is the same token in both paths and shares the same model parameters, π_θ is required to simultaneously increase and decrease their predicted occurrence probabilities during optimization. As a result, their gradient signals may cancel each other out in the parameter space. Furthermore, when their contexts are similar or even identical, the gradient directions tend to be highly aligned, with almost identical magnitudes. Specifically:

$$\nabla_\theta \log P_\theta(y^{(c)}|x, y_w^{(<i)}) \approx \nabla_\theta \log P_\theta(y^{(c)}|x, y_l^{(<j)}), \tag{6}$$

their contribution to the overall gradient of the loss function approaches zero: $\nabla_\theta f(\pi_\theta) \approx 0$

This implies that regardless of whether the contexts are entirely identical, when the same token appears in both the preferred and non-preferred answers, the optimization objective of DPO, although aiming to widen the probability gap between the preferred and non-preferred paths, struggles to effectively update the parameters at these common token positions. This optimization conflict not only could reduce convergence efficiency but also would weaken the model's ability to distinguish between the preferred and non-preferred paths in similar contexts, thereby suppressing the contrastive learning effect of DPO.

2.3 Identifying Ambiguous Tokens

To mitigate the negative impact of ambiguous vocabulary on preference alignment training, we propose a concise and effective method - AAO. This approach can identify ambiguous tokens in preference answers, encouraging LLMs to focus more on the core tokens that genuinely reflect human preferences. Specifically, we first encode each token in the answers into corresponding embedding vectors using the language model's embedding layer. To determine whether a token in one answer has semantically similar or identical tokens in the other answer, we compute the cosine similarity between each token embedding in the preferred answer and all token embeddings in the rejected answer, and vice versa. We then record the maximum similarity score for each token, formulated as:

$$S_\varepsilon^{(i)} = \max_{j \in K_{\neg\varepsilon}} \left(\frac{\mathcal{F}(e_\varepsilon^{(i)}, e_{\neg\varepsilon}^{(j)}) - \min_{k \in K_{\neg\varepsilon}} \mathcal{F}(e_\varepsilon^{(i)}, e_{\neg\varepsilon}^{(k)})}{\max_{k \in K_{\neg\varepsilon}} \mathcal{F}(e_\varepsilon^{(i)}, e_{\neg\varepsilon}^{(k)}) - \min_{k \in K_{\neg\varepsilon}} \mathcal{F}(e_\varepsilon^{(i)}, e_{\neg\varepsilon}^{(k)})} \right), \tag{7}$$

where $S_\varepsilon^{(i)}$ denotes the normalized similarity score of the i-th token in answer y_ε, with $\varepsilon \in \{w, l\}$ corresponding to the preferred and rejected answers, respectively. K_ε is the number of tokens in the answer, and $\mathcal{F}(\cdot)$ represents the cosine similarity function between two embedding vectors. To enhance the comparability of similarity scores across different tokens, we perform min-max normalization over all cross-answer token similarities and take the normalized maximum similarity as the final score for each token. Based on this score, we introduce a threshold to distinguish different categories of tokens. The specific threshold-setting strategy is detailed in Section 2.4. This approach enables the model to effectively identify and mitigate the influence of ambiguous tokens during training, thereby improving its ability to capture genuine human preference signals and enhancing preference alignment performance.

2.4 Adaptive Thresholds

Due to significant differences in semantic similarity between preferred answers across different tasks, and even among different queries, it is challenging to manually set fixed thresholds that can adapt to these varying semantic environments. To address this issue, we design a dynamic threshold adjustment mechanism that better accommodates the diverse semantics of different tasks. Specifically, we introduce a lightweight linear layer to automatically output the thresholds, formulated as follows:

$$a, b = AW(P(y|x, \theta)), \tag{8}$$

where $P(y|x, \theta)$ represents the output logits of the model, and AW is a multi-layer perceptron (MLP). The values of parameters a and b are constrained within the range $[0, 1]$. To ensure validity, we introduce a clipping operation during computation to guarantee that $a > b$ is always satisfied. This adaptive threshold module is designed to be lightweight and efficient, introducing minimal additional computational overhead while maintaining plug-and-play capability. Experimental results demonstrate that the impact of this module on time consumption is negligible (see Section 3.4).

To further optimize the threshold outputs of the linear layer, we design an additional loss function to assist in training, enabling the model to better learn adaptive thresholds for different tasks. This loss function consists of two main components: Fine-grained Contrastive Suppression Loss and Preference Reward Enhancement Loss.

Fine-grained contrastive suppression loss. To enable the reweighted generation to more precisely extract key tokens from both preferred and non-preferred data, we design a Fine-grained Contrastive Suppression Loss. Our goal is to significantly reduce the interference of background tokens and enhance the attention to key tokens, thereby optimizing the feature representation of preferred and non-preferred data more effectively.

Specifically, we define two types of similarity matrices:

$$\begin{aligned} S_{\text{pref}} &= \frac{\sum_{i=1}^{T_p} \sum_{j=1}^{T_d} \cos\left(E_{\text{pref},i} \cdot w_{\text{pref,i}}, E_{\text{dis},j} \cdot w_{\text{dis},i}\right)}{T_p \cdot T_d}, \\ S_{\text{dis}} &= \frac{\sum_{i=1}^{T_d} \sum_{j=1}^{T_p} \cos\left(E_{\text{dis},j} \cdot w_{\text{dis},i}, E_{\text{pref},i} \cdot w_{\text{pref,i}}\right)}{T_d \cdot T_p}, \end{aligned} \tag{9}$$

where T_p and T_d represent the number of tokens in the preferred and non-preferred data, respectively. E_{pref} and E_{dis} are the feature representations of the preferred and non-preferred data, while w_{pref} and w_{dis} are the corresponding weighting coefficients. We measure the average similarity between each token in the preferred data and all tokens in the non-preferred data, and vice versa, to quantify the feature differences between the two.

Based on this fine-grained comparison, we formulate the contrastive suppression loss as follows:

$$\mathcal{L}_{\text{contrastive}} = S_{\text{pref}} + S_{\text{dis}}. \tag{10}$$

The optimization objective of this loss function is to maximize the difference between the preferred and non-preferred data in the high-dimensional feature space, thereby enhancing the model's ability to capture preference expression effectively.

Preference reward enhancement loss. To further improve the model's performance on preferred samples (chosen), we introduce a Preference Reward Enhancement Loss. Unlike traditional contrastive learning, which focuses solely on the differences between positive and negative samples, our approach not only aims to increase the distinction between chosen and rejected samples but also seeks to significantly enhance the log-probability of chosen samples.

Specifically, given the model's output probability distribution $P(y|x)$, we calculate the sum of log-probabilities over all time steps for the preferred samples as the reward metric:

$$R_{\text{chosen}} = \sum_{t=1}^{T} \log P(y_t|x, \theta), \tag{11}$$

where y_t represents the output of the preferred data (chosen) at time step t, and θ denotes the model parameters. Under the adaptive weighting mechanism, the influence of background tokens is suppressed while the significance of key tokens is magnified. This adjustment enables the model to focus more on high-information regions during the generation of preferred answers, thereby increasing the log-probability.

To further strengthen this mechanism, we introduce an optimization objective for R_{chosen} in the loss function:

$$\mathcal{L}\text{reward} = -\mathbb{E}[R_{\text{chosen}}]. \tag{12}$$

This optimization objective implies that during each training iteration, the model is encouraged to enhance the log-probability of chosen samples, thereby amplifying the distinction between preferred and non-preferred samples in contrastive learning.

To summarize, the complete form of the auxiliary loss function is expressed as follows:

$$\mathcal{L}_{\text{auxiliaryloss}} = \mathcal{L}_{\text{contrastive}} + \mathcal{L}_{\text{reward}}. \tag{13}$$

2.5 Re-weighting Strategies

In this study, we introduce two similarity thresholds, a and b (with $a > b$), to distinguish tokens based on their semantic roles. Specifically, tokens with similarity scores greater than a are classified as ambiguous tokens, which represent highly redundant tokens that may introduce confusion during training. Tokens with similarity scores below b are treated as key tokens, indicating core words that are highly discriminative and more likely to reflect true human preferences. Tokens with similarity scores between b and a are considered transitional tokens, reflecting an intermediate semantic state between ambiguous and key tokens.

After identifying these three token categories, we design a targeted reweighting strategy to suppress the influence of ambiguous noise and enhance the model's focus on key semantics during preference alignment training. The strategy is as follows:

Suppressing ambiguous tokens. For ambiguous tokens, we reduce their contribution to the training loss by down-weighting their importance based on their similarity score. As the similarity approaches 1, the weight decreases non-linearly, diminishing their impact during optimization. The weight is computed as:

$$w_{\varepsilon}^{(i)} = (1 - S_{\varepsilon}^{(i)})^2. \tag{14}$$

This effectively reduces the influence of high-redundancy tokens in the optimization process.

Emphasizing key tokens. Since key tokens are more likely to carry genuine preference signals, we assign them higher training weights to encourage the model to focus on these crucial semantic units. The weighting function is defined as:

$$w_{\varepsilon}^{(i)} = 1 + S_{\varepsilon}^{(i)}. \tag{15}$$

This enhances the contribution of key tokens to parameter updates while maintaining smooth gradients.

Preserving transitional tokens. Transitional tokens have semantic similarity between ambiguous and key tokens, acting as semantic bridges to maintain information flow. During training, they are assigned a fixed weight of 1 without adjustment for several reasons: they carry auxiliary semantic details crucial for context completeness; reweighting could distort their natural distribution; keeping the weight stable balances the weakening of ambiguous tokens and emphasis on key tokens, enhancing training stability; and empirical results show limited benefits from adjusting their weights(see Section 3.4). Thus, setting transitional tokens' weight to 1 is a reasonable and effective choice that simplifies training while ensuring stable performance.

In summary, our final weighting formula is as follows:

$$w_{\varepsilon}^{(i)} = \begin{cases} (1 - S_{\varepsilon}^{(i)})^2 & \text{if } S_{\varepsilon}^{(i)} > a \\ 1 & \text{if } a < S_{\varepsilon}^{(i)} < b \\ 1 + S_{\varepsilon}^{(i)} & \text{if } S_{\varepsilon}^{(i)} < b \end{cases}. \tag{16}$$

However, during the actual training process, traditional threshold-based decisions do not support backpropagation, resulting in ineffective parameter updates for the adaptive model. To address this issue, we redesigned an alternative formulation to overcome the limitations of non-differentiable thresholds. The new formulation is expressed as follows:

$$\begin{aligned} w_\varepsilon^{(i)} &= \frac{(1-S_\varepsilon^{(i)})^2}{1+e^{-\alpha\cdot(S_\varepsilon^{(i)}-a)}} + \frac{(1+S_\varepsilon^{(i)})}{1+e^{-\alpha\cdot(b-S_\varepsilon^{(i)})}} \\ &+ (1 - \frac{1}{1+e^{-\alpha\cdot(S_\varepsilon^{(i)}-a)}} - \frac{1}{1+e^{-\alpha\cdot(b-S_\varepsilon^{(i)})}}), \end{aligned} \tag{17}$$

where α represents the fitting weight. When $\alpha = 200$, this parameter effectively fits Equation (16). Hence, in the subsequent experiments, we consistently set a to 200 to ensure result consistency.

Finally, we apply these weights to the logits of the final output from the LLMs, enabling the subsequent weighted training process.

3 Experiments and Results

3.1 Models, Datasets, and Evaluation Metrics

We evaluate the performance of the method from two aspects: open-domain instruction compliance benchmarks and safety alignment. During the preference optimization process, two types of models, Llama-3.1-8B (Grattafiori et al., 2024) and Mistral-7B (Jiang et al., 2024), are evaluated in both the base setting and the instruction setting. This section aims to investigate the performance of the AAO method compared to other preference optimization approaches under different experimental conditions.

Openended instruction-following benchmarks. In the base setting, we follow Zephyr's training procedure (Tunstall et al., 2023) by first training a base model on the UltraChat-200k dataset (Ding et al., 2023) to obtain an SFT model. Starting from this model, preference optimization is then performed on the UltraFeedback dataset (Cui et al., 2023). In the instruction setting, off-the-shelf instruction-tuned models are used as the SFT models; these models have undergone more extensive instruction tuning and are more powerful and robust than the SFT models in the base setting. Preference optimization is also conducted on the UltraFeedback dataset. We evaluate method performance using three widely adopted open-domain instruction compliance benchmarks: MT-Bench (Zheng et al., 2023), AlpacaEval 2 (Li et al., 2023b), and Arena-Hard (Li et al., 2024). For detailed settings, please see Table 1.

	Baseline	Judge Model	Metric
AlpacaEval 2	GPT-4 Turbo	GPT-4.1	LC & raw win rate
Arena-Hard	GPT-4-0314	GPT-4o	Win rate
MT-Bench	-	GPT-4o	Rating of 1-10

Table 1: Evaluation details for AlpacaEval 2, Arena-Hard, and MT-Bench. The baseline model refers to the model compared against.

Safety alignment evaluation. In the base setting, we train a base model on the Alpaca dataset (Taori et al., 2023) to obtain an SFT model, followed by preference optimization on the Anthropic-HH dataset (Bai et al., 2022a). In the instruction setting, off-the-shelf instruction-tuned models are used as SFT models, and preference optimization is likewise performed on the Anthropic-HH dataset. To assess harmlessness, we generated responses using aligned LLM on a mixed dataset of AdvBench (Zou et al., 2023) and JailbreakBench (Chao et al., 2024), and used Llama-Guard (Inan et al., 2023) to determine the safety of the responses.

3.2 Baseline Methods and Training Settings

We conducted comparative experiments to evaluate the proposed method against various baseline alignment methods, including sequence-level approaches such as DPO (Rafailov et al., 2023), IPO (Azar et al., 2024), KTO (Ethayarajh et al., 2024a), and SimPO (Meng et al., 2024), as well as token-level methods such as TDPO (Zeng et al., 2024), RTO (Zhong et al., 2024), and TIS-DPO (Liu et al., 2024). In addition, we introduced a randomly weighted method, DPO-Random, to further validate the effectiveness of our proposed weighting strategy. All baseline methods use the hyperparameter settings provided in their original papers. For our method, the learning rate is set to 5e-7, the batch size to 16, and training is performed for one epoch using the AdamW optimizer.

3.3 Main results

Table 2 presents our main experimental results. Despite its simplicity, our method achieves significant improvements across all metrics. On the LLaMA-3.1-8B-Base model, it raises the WR of AlpacaEval2 to 40.23%, 7.2 points higher than the second-best method. In the Arena-Hard benchmark, it further boosts performance to 41%, validating its effectiveness in enhancing generalization. Although MT-Bench is widely used, it shows weak

Method	AlpacaEval2		MT-Bench	Arena-Hard	Llama-Guard	AlpacaEval2		MT-Bench	Arena-Hard	Llama-Guard
	LC(%)↑	WR(%)↑	Avg.↑	WR(%)↑	Harm.(%)↑	LC(%)↑	WR(%)↑	Avg.↑	WR(%)↑	Harm.(%)↑
	Llama3.1-8B-Base					Llama3.1-8B-Instruct				
DPO	22.45	31.30	7.36	26.0	83.06	45.87	44.34	7.73	29.6	96.12
IPO	24.38	24.90	7.41	25.1	82.14	45.43	44.56	7.82	28.3	97.42
KTO	25.79	24.79	7.41	25.8	83.42	43.86	42.00	7.86	26.8	96.52
SimPO	27.45	33.03	7.47	30.6	84.15	47.50	43.64	7.91	33.5	97.85
TDPO	23.34	26.45	7.22	27.2	86.54	46.56	43.21	7.75	26.4	97.04
RTO	24.43	25.84	7.34	26.7	85.71	46.84	41.98	7.81	30.4	96.99
TIS-DPO	26.84	31.47	7.40	27.8	87.87	47.04	43.99	7.86	30.1	97.54
DPO-Random	19.07	33.97	7.11	25.4	83.42	36.36	35.10	7.33	27.9	97.32
AAO	**28.36**	**40.23**	**7.52**	**41.0**	**91.29**	**48.11**	**44.89**	**7.92**	**42.7**	**98.42**
	Mistral-7B-Base					Mistral-7B-Instruct				
DPO	20.45	17.55	7.16	8.1	95.48	32.62	28.70	7.51	17.0	99.14
IPO	17.13	12.13	7.33	7.5	96.34	31.46	30.31	7.56	16.8	99.42
KTO	16.49	09.32	7.31	8.2	95.04	34.65	30.90	7.64	18.6	99.21
SimPO	24.40	22.80	7.34	9.7	96.56	35.93	34.48	7.62	22.6	99.35
TDPO	21.47	18.56	7.18	7.9	94.55	35.34	30.55	7.52	18.3	97.97
RTO	20.40	16.77	7.26	8.1	94.36	34.49	32.52	7.55	17.9	98.76
TIS-DPO	21.57	18.84	7.25	8.2	96.74	34.54	33.51	7.58	18.0	97.35
DPO-Random	17.71	11.22	7.14	8.6	95.52	30.14	28.45	7.50	16.8	98.42
AAO	**25.01**	**23.45**	**7.36**	**13.6**	**97.41**	**36.66**	**38.54**	**7.65**	**24.2**	**99.67**

Table 2: AlpacaEval 2, Arena-Hard, MT-Bench and Llama-Guard results under the four settings. LC, WR, and Harm. represent length-controlled, raw win rate, and harmless response rate, respectively.

differentiation across methods due to its limited data size and single-instance scoring (Meng et al., 2024; Li et al., 2024). By contrast, AlpacaEval2 and Arena-Hard provide more reliable assessments. Even so, our method still achieves top performance across models and tasks, demonstrating strong robustness and adaptability. For safety alignment, our method scores 91.29% on LLaMA-3.1-8B-Base, 3.42 points above the second-best, indicating better distinction between preferred and non-preferred data.

3.4 Ablation Study

Analysis of weighting function. To validate the effectiveness of the proposed weighting strategy, we designed multiple weighting curves with different trends for comparative experiments. Among them, Functions (4) and (5) perform differentiated weighting operations on transitional tokens based on our strategy. The specific trends of each weighting curve are illustrated in Figure 2, and the corresponding experimental results are presented in Table 3. The experimental results clearly demonstrate that the proposed adaptive weighting strategy exhibits significant advantages across all metrics, fully proving its superiority. Moreover, when the weights of transitional tokens are adjusted, the model's training process is significantly affected, further validating the correctness of our hypothesis. Notably, the design of most weighting curves effectively optimizes the training performance of

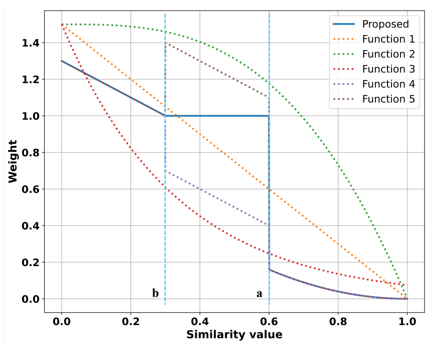

Figure 2: Images of different weighted curves. In our approach, the thresholds a and b are decided by LLM itself during training.

Method	AlpacaEval2		MT-Bench	Arena-Hard
	LC(%)↑	WR(%)↑	Avg.↑	WR(%)↑
Function 1	22.68	28.52	6.94	29.45
Function 2	23.74	32.38	7.14	32.44
Function 3	18.56	14.51	6.87	24.69
Function 4	27.98	37.24	7.42	38.84
Function 5	27.32	35.17	7.47	37.45
Proposed	**28.36**	**40.23**	**7.52**	**41.00**

Table 3: Results under different weighting methods.

DPO, which also reveals that the ambiguous phenomenon we proposed is both reasonable and existent. It should be noted that the choice of weighting curves remains an open question. In this study, we demonstrated the significant optimization effects of the proposed weighting curves, and future research could further explore more weighting strategies to enhance model performance.

Analysis of the auxiliary loss function. We

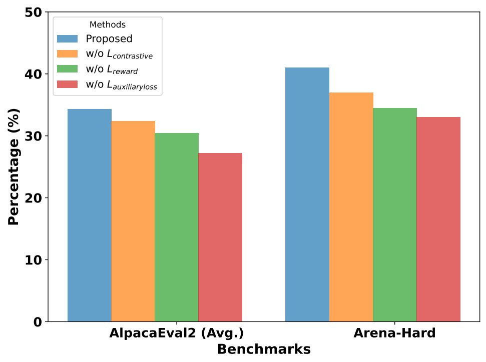

Figure 3: Effect of auxiliary losses on experimental results.

	Extra parameter size	Training time
DPO	-	77.81 mins
AAO	62.50MB	78.12 mins (0.4% ↑)

Table 4: Comparison of training time and additional parameters.

further explored the impact of each component in the auxiliary loss function on the final results, as illustrated in Figure 3. The results indicate that different components exhibit varying degrees of influence on the optimization of the adaptive module. When these components work synergistically, the model achieves optimal performance, validating the effectiveness and rationality of our design.

Analysis of time consumption. We conducted alignment experiments based on the Mistral-7B-Base model on the UltraFeedback dataset and compared the time overhead between DPO and AAO during the training process. The experimental results are presented in Table 4. The results indicate that our proposed adaptive model only consumes an additional 62.50 MB of storage space while not significantly increasing the training time, demonstrating its efficiency and lightweight design.

Squeeze effect. We further validated the effectiveness of AAO in alleviating the "squeezing effect" observed in DPO (Ren and Sutherland, 2024). As illustrated in Figure 4, during the training process, DPO significantly reduces the confidence of the highest-probability token while correspondingly increasing the probabilities of other tokens, clearly revealing the presence of the "squeezing effect." In contrast, our method effectively mitigates this phenomenon, further indicating that the cause of the "squeezing effect" is not solely attributed to sequence-level semantic similarity but also involves the handling of internal ambiguous labels.

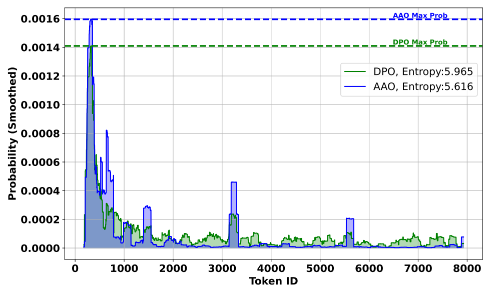

Figure 4: AAO mitigates the squeeze effect of DPO.

4 Related Work

RLHF has become a leading approach for aligning LLMs with human values, typically involving three stages: supervised fine-tuning, reward model training, and policy optimization using algorithms like PPO, GRPO, and REINFORCE++ (Ouyang et al., 2022; Casper et al., 2023; Achiam et al., 2023). Despite its effectiveness, RLHF's reliance on multiple models and LLM sampling increases complexity (Zhao et al., 2023; Casper et al., 2023). To address this, efficient offline alternatives like DPO have emerged, which transfer preference knowledge using both preferred and non-preferred responses without explicit reward models (Rafailov et al., 2023; Xu et al., 2024; Meng et al., 2024).

Recent studies indicate DPO optimizes LLMs by considering whole-response preferences, ignoring token-level importance, which constrains its performance (Zeng et al., 2024; Liu et al., 2024; Gu et al., 2025). To address this, our work explores the impact of token importance in DPO training, observing that semantically similar tokens in positive and negative examples may cause confusion. To solve this, we introduce AAO, an approach that allows the model to re-weight tokens automatically, mitigating confusion and enhancing DPO performance. Our analysis further uncovers that semantic confusion may contribute to the "squeeze effect" identified in (Ren and Sutherland, 2024).

5 Conclusion

In this paper, we theoretically and empirically identify potential ambiguity issues in DPO. To address this problem, we propose AAO, a simple yet effective approach that automatically re-weights background content based on semantic similarity. Extensive experiments demonstrate that AAO consistently outperforms existing methods across various training setups, validating its effectiveness.

6 limitations

Rigorous theoretical analysis. Although AAO has achieved practical success and is intuitively motivated, a more rigorous theoretical and experimental analysis is still necessary to fully understand the impact of ambiguous content during DPO training. Ideally, this can be accomplished by tracking the gradient dynamics associated with ambiguous content and combining these observations with theoretical derivations, in order to obtain more definitive conclusions and insights. We will leave this aspect for future work, aiming to gain a deeper understanding and provide a more quantitative analysis of the impact of ambiguous tokens during training.

Design of re-weighting strategy. In this study, we propose a piecewise similarity-based reweighting curve, designed based on empirical assumptions, to mitigate the adverse effects of ambiguous tokens. Our approach outperforms other curve designs and achieves state-of-the-art results. However, it may not be optimal; future work could explore enabling the model to automatically fit such curves or investigate novel mapping strategies.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. 2024. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pages 4447–4455. PMLR.

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, and 12 others. 2022a. Training a helpful and harmless assistant with reinforcement learning from human feedback. *Preprint*, arXiv:2204.05862.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, and 1 others. 2022b. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, and 1 others. 2023. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*.

Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J Pappas, Florian Tramer, and 1 others. 2024. Jailbreakbench: An open robustness benchmark for jailbreaking large language models. *arXiv preprint arXiv:2404.01318*.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

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.

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*.

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

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

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Yuzhe Gu, Wenwei Zhang, Chengqi Lyu, Dahua Lin, and Kai Chen. 2025. Mask-dpo: Generalizable fine-grained factuality alignment of llms. *arXiv preprint arXiv:2503.02846*.

Jiwoo Hong, Noah Lee, and James Thorne. 2024. Orpo: Monolithic preference optimization without reference model. *arXiv preprint arXiv:2403.07691*.

Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, and 1 others. 2023. Llama guard: Llm-based input-output safeguard for human-ai conversations. *arXiv preprint arXiv:2312.06674*.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, and 1 others. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Banghua Zhu, Joseph E Gonzalez, and Ion Stoica. 2024. From live data to high-quality benchmarks: The arena-hard pipeline. *lmsys Blog.(Apr. 19, 2024),[Online]. Available: https://lmsys.org/blog/2024-04-19-arena-hard/(visited on 08/04/2024)*.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023a. Alpacaeval: An automatic evaluator of instruction-following models.

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

Weibin Liao, Xu Chu, and Yasha Wang. 2024. Tpo: Aligning large language models with multi-branch & multi-step preference trees. *arXiv preprint arXiv:2410.12854*.

Zhenghao Lin, Zhibin Gou, Yeyun Gong, Xiao Liu, Yelong Shen, Ruochen Xu, Chen Lin, Yujiu Yang, Jian Jiao, Nan Duan, and 1 others. 2024. Rho-1: Not all tokens are what you need. *arXiv preprint arXiv:2404.07965*.

Aiwei Liu, Haoping Bai, Zhiyun Lu, Yanchao Sun, Xiang Kong, Simon Wang, Jiulong Shan, Albin Madappally Jose, Xiaojiang Liu, Lijie Wen, and 1 others. 2024. Tis-dpo: Token-level importance sampling for direct preference optimization with estimated weights. *arXiv preprint arXiv:2410.04350*.

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

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. 2024. Disentangling length from quality in direct preference optimization. *arXiv preprint arXiv:2403.19159*.

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. *Advances in Neural Information Processing Systems*, 36:53728–53741.

Yi Ren and Danica J Sutherland. 2024. Learning dynamics of llm finetuning. *arXiv preprint arXiv:2407.10490*.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021.

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

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro Von Werra, Clémentine Fourrier, Nathan Habib, and 1 others. 2023. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*.

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. *arXiv preprint arXiv:2401.08417*.

Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. 2023. Rrhf: Rank responses to align language models with human feedback. *Advances in Neural Information Processing Systems*, 36:10935–10950.

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

Hanyang Zhao, Genta Indra Winata, Anirban Das, Shi-Xiong Zhang, David D Yao, Wenpin Tang, and Sambit Sahu. 2024. Rainbowpo: A unified framework for combining improvements in preference optimization. *arXiv preprint arXiv:2410.04203*.

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

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

Han Zhong, Zikang Shan, Guhao Feng, Wei Xiong, Xinle Cheng, Li Zhao, Di He, Jiang Bian, and Liwei Wang. 2024. Dpo meets ppo: Reinforced token optimization for rlhf. *arXiv preprint arXiv:2404.18922*.

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

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*.