

Debiasing Online Preference Learning via Preference Feature Preservation

Dongyoung Kim¹, Jinsung Yoon², Jinwoo Shin¹, Jaehyung Kim³

¹KAIST, ²Google Cloud AI Research, ³Yonsei University
kingdy2002@kaist.ac.kr, jaehyungk@yonsei.ac.kr

Abstract

Recent preference learning frameworks for large language models (LLMs) simplify human preferences with binary pairwise comparisons and scalar rewards. This simplification could make LLMs' responses biased to mostly preferred features, and would be exacerbated during the iterations of online preference learning steps. To address these challenges, we propose a novel framework coined PFP (Preference Feature Preservation). The key idea of PFP is maintaining the distribution of human preference features and utilizing such rich signals throughout the online preference learning process. Specifically, PFP first extract preference features from offline pairwise human preference data and trains a feature classifier. Then, using trained classifier and the distribution preserving optimization, PFP maps appropriate preference features for a new input instruction during online learning. Lastly, PFP trains LLM using the existing preference learning method, by incorporating the preference feature into system prompts and enabling LLM to explicitly handle various human preferences. Our experiments demonstrate that PFP successfully mitigates the bias in preference features during online learning, and hence achieves superior performance compared to previous preference learning methods on standard benchmarks to evaluate LLM alignment.¹

1 Introduction

Aligning large language models (LLMs) using human feedback, particularly by learning from human preferences, yields remarkable successes in various NLP tasks and real-world applications such as coding assistants and chatbots (Anthropic, 2024; Dubey et al., 2024; OpenAI, 2024b; Team et al., 2023). To improve the alignment of LLMs, various preference learning algorithms, such as Reinforcement Learning from Human Feedback (RLHF)

(Ouyang et al., 2022) and Direct Preference Optimization (DPO) (Rafailov et al., 2023), have been explored. A common assumption across these works is that human preference is provided in a binary pair-wise comparison (Ziegler et al., 2019; Hong et al., 2024). This approach enables easy modeling of human preference using the scalar reward such as the Bradley-Terry (BT) model (Bradley and Terry, 1952).

However, human preference is determined by several underlying features (Li et al., 2024; Oh et al., 2024), and hence such simplification has critical limitations and fails to capture the complexity of human preferences. For example, even though human preference can be varied depending on the preference feature, the most dominant preference feature would be only considered to determine the binary human preference label. This issue becomes even more problematic in *online preference learning* scenarios, which progressively improves the alignment of LLMs by iterating the generation of preference data and learning from them (Xiong et al., 2024; Wu et al., 2024; Rosset et al., 2024). During online preference learning, LLM will generate responses biased toward specific preference features, and the preference annotators, such as the external reward model (Jiang et al., 2023b), will provide positive feedback on this. As such iterations go on, the bias of LLM accumulates (see Fig. 1(a)), and hence it results in the reduced diversity and quality of LLM's responses.

Contribution. To address these challenges, we propose a novel online preference learning framework called PFP (Preference Feature Preservation). Our approach is to ensure that the distribution of preference features remains consistent throughout the online preference learning process. Here, the key idea is to explicitly extract preference features of each input instruction and handle them using system prompts of LLMs (see Fig. 1(b)); it enables LLMs to generate and learn preference data

¹<https://github.com/kingdy2002/PFP>

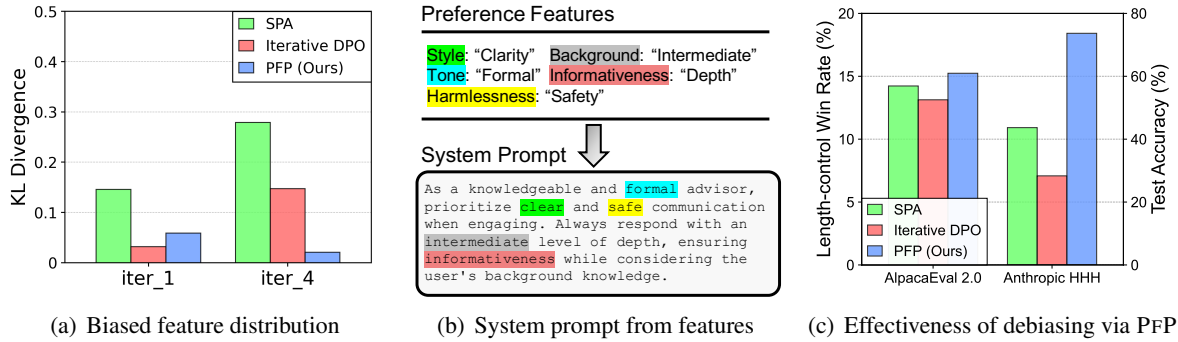


Figure 1: **Motivation for debiasing online preference learning.** (a) For existing online preference learning methods (SPA, Iterative DPO), the preference feature distribution of trained models, obtained by inversely asking GPT-4o, progressively diverges from the original preference distribution, captured by the initial DPO model. (b) To preserve the distribution, we propose to map each input instruction with the specific preference features and then convert it into the system prompt to enable LLM to explicitly handle preference features and preserve them. (c) We demonstrate that our approach, PFP, achieves strong performance not only on general preference benchmarks such as AlpacaEval2.0 but also on Anthropic-HHH, which incorporates critical metrics like harmlessness and honesty.

with intent. Specifically, PFP first estimates the initial distribution of preference features of the given human preference dataset, by inferring which features mainly determine binary human preferences. We then train a preference feature classifier, which maps each input instruction to appropriate preference features with additional optimization for the distribution preservation, during the online learning process. Finally, PFP trains LLM using the existing preference learning framework, by converting the mapped preference features of each generated data into the system prompts of LLMs.

We demonstrate the effectiveness of the proposed PFP by applying it to align recent open-sourced LLMs with the commonly used preference dataset, UltraFeedback (Cui et al., 2023). The experimental results demonstrate that PFP successfully prevents LLMs being biased to specific preference features during online learning, and results in the improved alignment in various aspects (Fig. 1(c)). For example, our framework achieves 7.66% increase in AlpacaEval 2.0 length-controlled win rate compared to the SFT model. Also, PFP achieves 2.11% larger increase compared to Iterative DPO, an online preference learning method with external reward model. In addition, unlike other baselines for online preference learning, PFP successfully improves harmlessness and honesty of the responses (Askell et al., 2021) through multiple iterations. More interestingly, PFP exhibits additional advantage that reduces the occurrence of length bias during online preference learning, despite not being specifically designed to address this. Overall, these results demonstrate that PFP is highly effective and practical for real-world appli-

cations, and underscoring the importance of debiasing to learn human preference for LLM alignment.

2 Related Works

LLM alignment with human preference. Aligning LLMs with human intentions and values using human feedback data now becomes a defacto standard to obtain well-performing LLMs (Ziegler et al., 2019; Ouyang et al., 2022). Typically, this feedback is collected by asking human annotators to compare two responses generated from the same input prompt and assign a binary preference label. One of the most widely adopted approaches is RLHF (Christiano et al., 2017; Stiennon et al., 2020), where a reward model is trained to model human preferences (Bradley and Terry, 1952), and LLM is then fine-tuned to optimize for this learned reward. To prevent issues such as reward over-optimization and model collapse, KL divergence regularization is commonly employed during this process. However, RLHF presents several challenges such as computational overheads and the training instability. To address these issues, alternative approaches have been extensively proposed (Rafailov et al., 2023; Zhao et al., 2023; Meng et al., 2024; Hong et al., 2024); for instance, DPO (Rafailov et al., 2023) eliminates the need for a separate reward model by deriving a training objective that is mathematically equivalent to RLHF.

Online preference learning. Existing preference learning methods can generally be categorized into two approaches depending on whether they use the fixed human preference dataset (*offline preference learning*, e.g., DPO) or progressively enlarge dataset from the iterations of sampling and labeling

(online preference learning, e.g., RLHF). While online methods typically achieve superior performance due to train with more data, they also demand more computations from sampling responses and labeling preferences. To address this challenge, recent work has focused on developing efficient batch-online preference learning methods, such as Iterative DPO (Xu et al., 2023; Xiong et al., 2024; Rosset et al., 2024; Wu et al., 2024; Calandriello et al., 2024). Iterative DPO generates thousands of responses in each iteration (batch) and constructs labeled preference datasets by judging the preference using the reward model (Jiang et al., 2023b). This dataset is then used to train LLMs with offline methods like DPO, and the iteration repeats, resulting in more efficient and stable alignment.

3 Preliminary

Let the LLM policy be denoted as π_θ , which can generate output sequence (i.e. response) y , given input sequences composed of *system prompt* s and *instruction* x , i.e., $y \sim \pi_\theta(s, x)$. Here, the system prompt s is usually considered to be fixed regardless of the input instruction x . For convenience, we assume that s is always included as the input of π_θ and hence omit s in the equations in the below parts. Next, we assume that we have the labeled preference dataset, $\mathcal{D} = \{(x, y_l, y_w)\}$, where y_l and y_w are the dis-preferred and preferred responses for the corresponding instruction x , respectively.

RLHF and DPO. To train π_θ with \mathcal{D} for the alignment, RLHF first introduces the reward model $r(x, y)$ which can convert human preference data into scalar values. Specifically, the reward model $r(x, y)$ is often modeled with the Bradley-Terry (BT) model (Bradley and Terry, 1952), and then it can yield the probability $p(y_w \succ y_l | x)$ that response y_w is preferred over y_l as follow:

$$p(y_w \succ y_l | x) = \frac{\exp(r(x, y_w))}{\exp(r(x, y_w)) + \exp(r(x, y_l))}.$$

As the optimal reward function $r(x, y)$ is not accessible, a parameterized reward model $r_\phi(x, y)$ is usually introduced by optimizing its parameters with the maximum-likelihood objective on the preference dataset. With this reward model, RLHF optimizes LLM π to maximize this reward with the additional regularization of the KL divergence between the current policy and the reference policies

(π_{ref}) to prevent reward over-optimization:

$$\begin{aligned} \mathcal{L}_{\text{RLHF}} = & -\mathbb{E}_{y \sim \pi_\theta, x \sim \rho} [r_\phi(x, y)] \\ & + \beta \text{D}_{\text{KL}}(\pi_\theta(y|x) \parallel \pi_{\text{ref}}(y|x)). \end{aligned}$$

To remove the necessity of the reward model in RLHF, DPO proposed a method that is mathematically equivalent to the original RLHF objective and can directly optimize the internal reward modeled by LLM π itself, by maximizing the weighted likelihood gap between y_w and y_l :

$$\begin{aligned} p_\theta(y_w \succ y_l | x) = & \sigma\left(\beta \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)}\right. \\ & \left. - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\right). \end{aligned} \quad (1)$$

$$\mathcal{L}_{\text{DPO}} = \mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} [-\log p_\theta(y_w \succ y_l | x)].$$

Online preference learning and SPA. In the online preference learning scenario, we have unlabeled instruction datasets $X_t = \{x\}$, $t = 1, \dots, T$ where $X_t \cap X_{t'} = \emptyset$ when $t \neq t'$. For t -th iteration, the preference dataset $\mathcal{D}_t = \{(x, y_l, y_w) | x \in X_t\}$ is constructed by (1) sampling two responses for each instruction $x \in X_t$ using LLM policy π_{t-1} from the previous iteration, i.e., $y_1, y_2 \sim \pi_{t-1}(x)$, and (2) judging the preference between them. Then, LLM policy π_t which is initialized with π_{t-1} is trained with \mathcal{D}_t using the existing preference learning method. One representative approach is Iterative DPO (Xu et al., 2023), where the external reward model is used for the preference judgments and π_t is trained with \mathcal{D}_t using DPO. Since choosing the proper reward model is non-trivial in our problem, we adopt SPA (Kim et al., 2025) as the online preference learning algorithm. SPA conducts preference labeling using the implicit reward derived from the DPO’s objective function (Eq. 1), unlike the other online preference learning methods using the external reward model:

$$\begin{aligned} (y_w, y_l) = & (y_1, y_2) \text{ if } p_{\theta_{t-1}}(y_1 \succ y_2 | x) > 0.5 \\ (y_w, y_l) = & (y_2, y_1) \text{ else} \end{aligned} \quad (2)$$

4 PFP: Debaised LLM Alignment via Preference Feature Preservation

In this section, we present PFP (Preference Feature Preservation) to align LLM by reducing the bias during online preference learning. Our main idea is to explicitly extract preference features of input instruction, and handle them using system prompts of LLM. The overview of PFP is presented in Fig. 2.

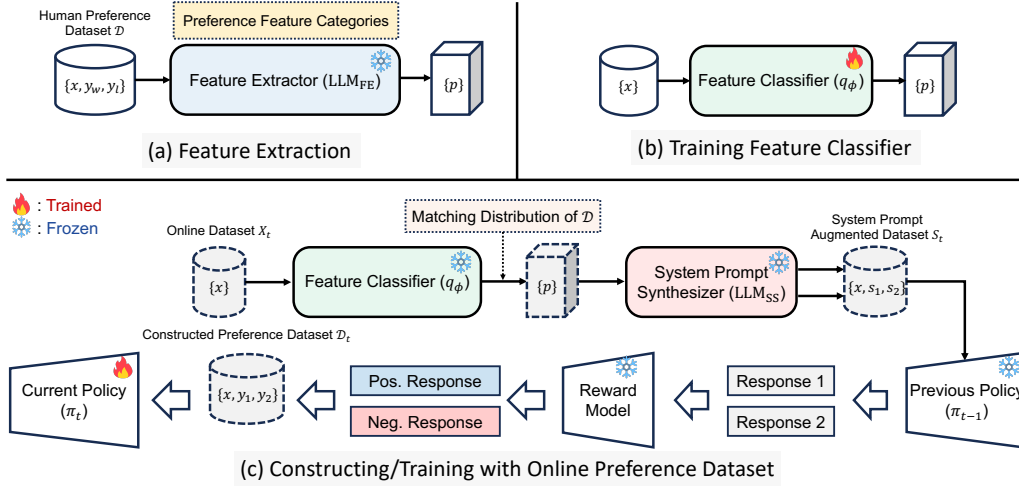


Figure 2: **Illustration of the proposed PFP framework.** (a) PFP first extracts the preference feature among the predefined categories for a given human preference dataset using an LLM-based feature extractor. (b) With the extracted features, PFP trains the feature classifier. (c) The trained feature classifier along with an additional distribution adjustment step assigns the preference feature for a new instruction in a online data. Then, the LLM-based system prompt synthesizer converts it into two system prompts, where each system prompt is used to sample the separate response. Then, the labeled preference dataset is constructed and the current policy LLM is trained.

4.1 Extracting preference feature from binary human preference data

We first assume that some underlying keys features mainly determine the human preference between responses for a given input prompt; we call them as *preference features*. Following Lee et al. (2024), we define a five-dimensional preference feature where each dimension captures different domain of preference: *style, tone, harmlessness, background knowledge, and informativeness*. Then, each dimension contains five sub-features; for example, *style* dimension consists of following five sub-features: *clarity, conciseness, format, vividness, and consistency* (see full description in Table 5). Under this definition, we extract the preference features of the pairwise offline human preference data \mathcal{D} using the feature extractor. We implement the feature extractor by prompting LLM such as GPT-4o (OpenAI, 2024b), to infer the likely preference features that led the annotators to provide a given preference label. Specifically, for the input instruction x and the two responses y_w and y_l , the feature extraction yields $\mathbf{p} = \text{LLM}_{\text{FE}}(x, y_l, y_w)$ where $\mathbf{p} = [p_1, \dots, p_5]$, where each p_i represents a one-hot label over the 5 sub-features for i -th dimension (i.e., $p_i \in [0, 1]^5$ and $\sum_{j=1}^5 p_i^j = 1$). Then, the extracted preference features are added to the human preference data \mathcal{D} and it yields $\mathcal{D}_{\text{FE}} = \{(\mathbf{p}, x, y_l, y_w)\}$.

4.2 Distribution preserved feature mapping

To preserve the feature distribution over each iteration of online preference learning, we first map

each instruction $x \in X_t$ used in online learning to the proper preference features. Then, the preference feature distribution can be preserved by explicitly incorporating the assigned features during response generation and preference judgment. Specifically, this process involves two key components: (a) learning feature classifier, and (b) adjusting assigned feature using relabeling technique.

Learning feature classifier. PFP introduces an auxiliary classifier q_ϕ to predict appropriate preference features for the given input instruction x . Specifically, q_ϕ is trained with a conventional supervised learning with cross-entropy loss, using the input instructions x and the extracted features p in \mathcal{D}_{FE} (i.e., sequence classification). After the training, q_ϕ can provide a probability distribution over preference features for a new instruction $x \in X_t$ that will be used in online learning. For each i -th dimension, a separate classifier q_{ϕ^i} is introduced where $q_{\phi^i}(x) = [0, 1]^5$ and $\sum q_{\phi^i}(x) = 1$.

Adjusted output prediction. However, due to the difficulty of the given task from limited data and long-tailed nature, the classifier’s prediction can be inaccurate and hence limited to preserve the feature distribution. To complement the classifier’s predictions be aligned with the distribution of human preferences, PFP adjusts the predicted probabilities by solving the optimization for this. Formally, for each i -th preference dimension, the human preference feature distribution is empirically derived from \mathcal{D}_{FE} , i.e., $P_i = \sum_{\mathbf{p} \in \mathcal{D}_{\text{FE}}} p_i / |\mathcal{D}_{\text{FE}}|$. Next, the output probabilities for all input instruc-

tions in X_t under q_{ϕ^i} is collected to measure the distribution, *i.e.*, $Q_i = \sum_{x \in X_t} q_{\phi^i}(x)/|X_t|$. Then, our goal is to find the adjusted output probability $\tilde{q}_i(x) \in [0, 1]^5$, $\sum \tilde{q}_i(x) = 1$ for $x \in X_t$ that yields the identical empirical distribution with P_k while minimizing the deviation from the original probability $q_{\phi^i}(x)$. This problem can be formulated as below optimization problem:

$$\begin{aligned} \min_q \text{CE}(q_{\phi^i}, q) \quad \text{s.t.} \quad & \forall x \in X_t : q(x) \in [0, 1]^5, \\ & \sum_{j=1}^5 q(x)_j = 1, \text{ and } \sum_{x \in X_t} q(x)/|X_t| = P_i. \end{aligned} \quad (3)$$

where CE is a cross-entropy. Following the previous works (Asano et al., 2020; Kim et al., 2020), we solve this problem via efficient Sinkhorn-Knopp algorithm (Cuturi, 2013). With $\tilde{q}_k(i)$ from solving Eq. 3 with q_{ϕ^i} , we sample the preference feature and augment the online dataset X_t , *i.e.*, $\tilde{p}_i \sim \tilde{q}_i(x)$ and $\tilde{X}_t = \{(\tilde{\mathbf{p}}, x) | x \in X_t, \tilde{\mathbf{p}} = [\tilde{p}_1, \dots, \tilde{p}_5]\}$.

4.3 Learn to handle preference features through system prompt

Synthesizing system prompt. We need to generate responses and judge the preference using the LLM policy π_{θ} conditioned on the given preference feature. However, it can be difficult as the preference features have the form of short words that are not suitable for LLM. To address this, we convert these discretized preference features into the system prompt, which is a natural language description about the preference feature. Then, we add it in front of the instruct x as the conventional system prompt. Specifically, the system prompt s is created through the system prompt synthesizer, which is implemented by prompting LLM that receives features as input and generates a system prompt, *i.e.*, $s \sim \text{LLM}_{\text{SS}}(\tilde{p})$. Then, we incorporate the generated system prompt into the online learning dataset, *i.e.*, $S_t = \{(s, x) | (\tilde{\mathbf{p}}, x) \in \tilde{X}_t\}$. We created the prompt for LLM_{SS} by modifying the prompt used in Lee et al. (2024) (see Appendix F). Using S_t , one can perform the existing online preference learning method, such as Iterative DPO. **System prompt sampling and scheduling.** While incorporating preference features into LLM using the system prompt enables LLM to understand and handle them better, we observe that conditioning specific system prompts could reduce the diversity between sampled responses. This reduced diversity makes preference judgment between them difficult

and consequently leads to decreased performance (see Table 3). To prevent this, we propose to augment the online learning data set X_t by sampling two system prompts, *i.e.*, $S_t = \{(s_1, s_2, x) | x \in X_t\}$ and $s_1, s_2 \sim \text{LLM}_{\text{SS}}(\tilde{p})$. Then, during data construction process, each system prompt is used to sample the different response, *i.e.*, $y_i \sim \pi_{t-1}(s_i, x)$ where $i = 1, 2$. Finally, using Eq. 1 and 2, we judge the preference between y_1 and y_2 with randomly chosen s between s_1 and s_2 , and construct the labeled dataset $\mathcal{D}_t = \{(s, x, y_l, y_w) | x \in X_t\}$.

In addition, to improve the effectiveness of online preference learning, we propose progressively increasing the training examples’ difficulty akin to curriculum learning (Bengio et al., 2009). To this end, we simply reduce the temperature used for system prompt sampling as the iteration increases, which reduces the diversity between two system prompts. We expect that it also reduces the distance between two responses y_1 and y_2 from online response sampling with π_{t-1} and S_t , *i.e.*, more difficult to learn; therefore, this approach improves the effectiveness of online preference learning by continuously increasing the difficulty of the task. We present full procedure of PFP in Algorithm 1.

5 Experiments

5.1 Setups

In this section, we first present our experimental setups. As denoted in Sec. 3, we adopt SPA framework (Kim et al., 2025) as our online preference learning algorithm for the experiments. SPA enables the effective alignment of LLMs with limited preference data and does not require the external reward model; SPA includes the process of using initial seed data to train and create the initial DPO model. Here, the initial DPO model acts as the base model as well as the reward model before the iterative learning process begins (Eq. 1 and 2).

Models. For the policy LLM, we utilize an open-source model supervised fine-tuned (SFT) on UltraChat data (Ding et al., 2023) based on the Mistral-7B-0.1v model (Jiang et al., 2023a), following the Zephyr recipe (Tunstall et al., 2023). For the feature classifier q_{ϕ} (Sec. 4.2), we employ DeBERTa-v3-large (He et al., 2023) as the backbone. We create five separate classifiers, one for each class of preference feature.

Datasets. For the initial labeled preference data, we use UltraFeedback dataset (Cui et al., 2023) which has been extensively used by prior works

(Snorkel, 2024; Rosset et al., 2024; Kim et al., 2025). Specifically, we sample 10K samples to construct a seed dataset. For PFP, the seed data would be taken feature extraction and system prompt synthesis processes, and the resulting data with added system prompts are used for initial DPO training and feature classifier training. Excluding seed data, we sample 4 datasets of 5K input prompts each, ensuring no overlap; these datasets are used to generate responses in each iteration of online learning. **Baselines.** We consider various preference learning baselines: *DPO* (Rafailov et al., 2023), *Iterative DPO* (Xiong et al., 2024), and *SPA* (Kim et al., 2025). All models under different baselines are initialized with the same SFT model. Iterative DPO, SPA, and PFP use the same online instruction datasets for each iteration. For the reward model in Iterative DPO, we employ PairRM (Jiang et al., 2023b), which is widely used in LLM alignment. While the initial DPO model was originally adopted as a base model only for PFP and SPA, we also consider using initial DPO as a base model in the case of Iterative DPO for a fair comparison. Specifically, we train initial DPO model using the seed dataset without mapped system prompts.

Evaluations. To evaluate trained models, we employ commonly used benchmarks in preference alignment research as follows. AlpacaEval 2.0 (Dubois et al., 2024) is designed to approximately evaluate human preference for instruction following, and calculates the win rate by comparing the response of GPT-4 (OpenAI, 2023) and the target model response by using GPT-4 as the evaluator. It is known that this benchmark reflects human preferences well, including a length-controlled win rate that reduces the impact of length bias. Next, MT-Bench (Zheng et al., 2023) is designed to evaluate more diverse capabilities of LLMs by utilizing GPT-4 to score the responses of the model under evaluation on a scale from 0 to 10. Lastly, we evaluate the performance at the Anthropic-HHH (Askill et al., 2021) which is designed to evaluate the alignment of LLMs with respect to three key attributes: Helpfulness, Honesty (*i.e.* accuracy), and Harmlessness. For each data consisting of a human query and two corresponding AI responses, we assess model alignment by measuring whether the internal preference (Eq. 1) is aligned with human labeled preference. In addition, to measure the debiasing effect on preference features, we extract the preference features from the responses generated for the test instructions in AlpacaEval

Table 1: Results with different variants of Mistral-7B-v0.1. The best scores are highlighted in **bold**.

Models	AlpacaEval 2.0		MT-Bench
	Len-control. Win Rate (%)	Avg. len (# chars)	Avg. Score (0-10)
SFT	7.58	901	6.34
DPO (W/o sys)	9.93	1409	6.34
DPO (W sys)	9.27	1135	6.61
SPA	14.23	2412	6.56
Iterative DPO	13.13	1709	6.53
PFP (Ours)	15.24	1187	6.88

2.0. Then, we use GPT-4o (OpenAI, 2024a) to infer the most prominent preference feature in each response. After obtaining the feature distribution, we measure how the KL divergence between this and the feature distribution of the responses of the initial model, which is trained on seed dataset.

Implementation details. We extract preference features of the seed preference data using GPT-4o. Here, we set the temperature as 0 and employ zero-shot chain-of-thought (CoT) prompting (Wei et al., 2022; Kojima et al., 2022). We train the feature classifiers using Adam optimizer (Kingma, 2015) with a learning rate of $1e-5$, a batch size of 32, over 5 epochs. We also use GPT-4o to synthesize system prompts, taking preference features as input. For double prompt sampling and scheduling (Sec. 4.3), the system prompts in the first iteration are generated with a temperature of 1.25, decreasing by 0.25 with each subsequent iteration. If scheduling is not applied, system prompts are generated with a temperature of 1. For subsequent iterations and the initial DPO, we set $\beta = 0.1$ and train for 1 epoch with a batch size of 32. This value is the same throughout PFP and SPA learning, but in the case of Iterative DPO, $\beta = 0.01$ was used during online learning. The learning rate of $5e-7$ is used with AdamW optimizer (Loshchilov et al., 2017). We employ a cosine learning rate scheduler with a 0.1 warm-up ratio. For PFP, Iterative DPO, and SPA, response sampling is performed twice per prompt with a temperature of 0.7. Unlike the original SPA, we remove self-refine step to reduce the number of hyperparameters. All the prompts used with GPT-4o are presented in Appendix F.

5.2 Main results

In Table 1, we present the performance of the models obtained after 4 iterations of online preference learning, including the results of baselines such

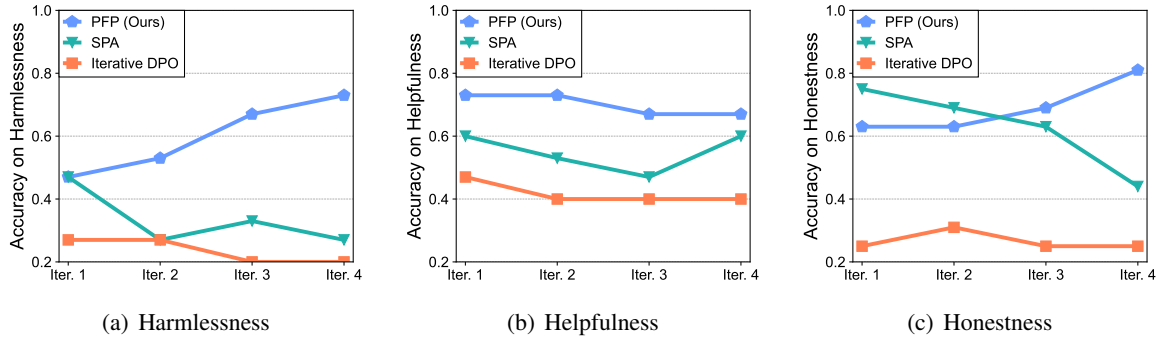


Figure 3: **Change of Accuracy on Anthropic-HHH.** Accuracy of the model’s internal preference (Eq. 1) on Anthropic-HHH. For the reference model for Eq. 1, the initial DPO model is used. As training progresses, PFP exhibits significant improvements in harmlessness and honesty, whereas other baselines show notable declines.

as used SFT and DPO trained on initial preference dataset. Notably, PFP achieves the higher performance than SPA (7.58 \rightarrow 14.23) and Iterative DPO (7.58 \rightarrow 13.13) with a performance improvement of (7.58 \rightarrow 15.24) based on AlpacaEval 2.0 length-controlled win rate. In MT-Bench, PFP also showed a large improvement (6.34 \rightarrow 6.88) compared to SPA (6.34 \rightarrow 6.56) and Iterative DPO (6.34 \rightarrow 6.53). These significant improvements in both AlpacaEval 2.0 or MT-Bench clearly affirm the overall enhancement in performance from PFP.

Next, in Fig. 3, we present the performance trajectory for each method on the Anthropic-HHH benchmark. In particular, PFP consistently maintains high performance in all categories evaluated during the learning process; PFP shows steady improvements in both harmlessness (0.47 \rightarrow 0.73) and honesty (0.63 \rightarrow 0.81). In contrast, Iterative DPO and SPA exhibit significant performance declines in these two metrics over the iterative training process. To verify whether these gains from PFP stems from preserving the initial preference distribution and mitigating the bias, We compare the KL divergence of the responses from each method and the preference distribution from the initial DPO model (see more details in Eq. 4). The results are presented in Fig. 1(a). In the case of Iterative DPO and SPA, the distribution diverges at the end, while in the case of PFP, the marginal change in distribution occurs as iteration progresses. This represents that the existing iterative improvement algorithm has bias at the feature level, and PFP sufficiently alleviates this. Additional experimental results including the effectiveness on LLaMA3-8B (Dubey et al., 2024) are presented in Appendix E.

5.3 Ablation study

To evaluate the effect of the feature labeling method (Sec. 4.2), we conduct an ablation study by remov-

Table 2: Results with iteratively trained models (from initial DPO) under different configurations of PFP. SP, CL, RL are abbreviations of system prompt, classifier label, and relabeling, respectively. When using only the system prompt, features are mapped randomly.

Components			AlpacaEval 2.0		MT-Bench
SP	CL	RL	Len-control. Win Rate (%)	Avg. len (# chars)	Avg. Score (0-10)
✓	✗	✗	11.58	1211	6.76
✓	✓	✗	12.52	1226	6.45
✓	✓	✓	15.24	1187	6.88

ing some components. Table 2 shows the corresponding experimental results after a total of 4 iterations. For the method in 1st row, the random feature is created by generating a preference feature regardless of the prompt. For the others, the preference features are sampled based on the probability of the feature classifier for each input instruction. Additionally, for the 3rd row, we relabel the probability of the features according to Eq. 3 to preserve the original human preference distribution. In the results, it is observe that the sampled feature through the classifier is solely not sufficient; the performance is increased (11.58 \rightarrow 12.52) on AlpacaEval 2.0. However, on MT-bench, the performance is decreased (6.76 \rightarrow 6.45). However, by applying relabeling technique, the performances on both benchmarks are successfully improved. Meanwhile, as shown in Fig. 4(a), the preference feature distribution is successfully preserved when applying both feature classifier and relabeling.

In addition, to evaluate the effect of the response sampling method (Sec. 4.3), we conduct additional ablation study by varying the usage of double system prompt sampling and scheduling. As shown in Table 3, the double system prompt sampling is effective to improve the performance on AlpacaEval

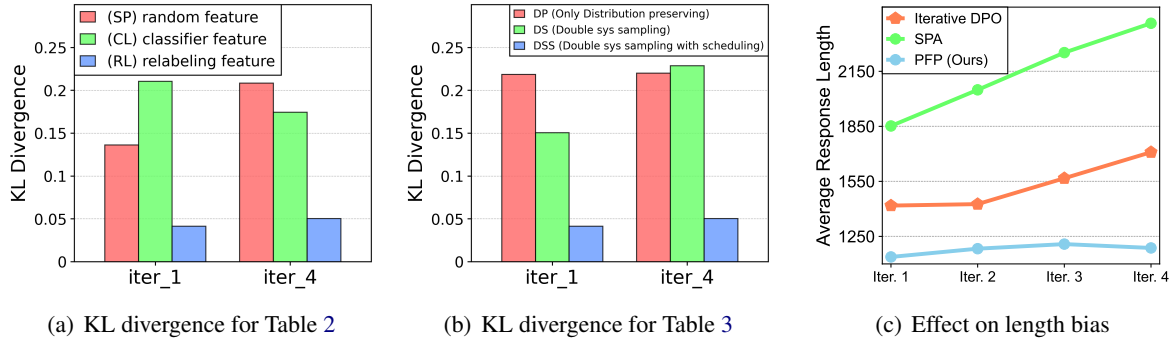


Figure 4: **Analyses.** (a,b) KL divergence with feature distribution from different methods for ablation, (c) Change of average response length on AlpacEval 2.0 with different methods.

Table 3: Results with iteratively trained models (from initial DPO) under different configurations of PFP. DP, DS, DSS are abbreviations of distribution preserving, double system prompt sampling, and double system prompt sampling with scheduling, respectively.

Components			AlpacEval 2.0		MT-Bench
DP	DS	DSS	Len-control. Win Rate (%)	Avg. len (# chars)	Avg. Score (0-10)
✓	✗	✗	10.79	1276	6.69
✓	✓	✗	12.30	1217	6.56
✓	✓	✓	15.24	1187	6.88

2.0. When scheduling is further applied, the improvement is enlarged, with AlpacEval 2.0 (12.30 \rightarrow 15.24) and MT-Bench (6.56 \rightarrow 6.88). Additionally, these components not only improve performance but also play a significant role in bias mitigation. As shown in Fig. 4(b), double system prompt sampling and scheduling are greatly effective to preserve the feature distribution. These results confirm that the proposed double system prompt sampling and scheduling are key factors to enhance performance by mitigating the bias.

5.4 Mitigation of Length Bias

From the previous experiments, we observe that PFP exhibits an unexpected additional advantages to mitigate *length bias* of LLM (see Fig. 4(c)), where aligned LLMs tend to generate and favor the longer responses (Park et al., 2024; Singhal et al., 2023). As it becomes critical problem by complicating the accurate assessment of LLM performance (Dubois et al., 2024; Wang et al., 2023), many works have proposed to tackle this problem. For example, the length penalty method works by heuristically subtracting a bias based on the length in the reward term (Dong et al., 2024). Alternatively, R-DPO approach (Park et al., 2024) adds additional length regularization into DPO objective.

Table 4: Results with iteratively trained models (from initial DPO) under different methods to mitigate length bias (length penalty and R-DPO). The best scores are highlighted in **bold**.

Methods	AlpacEval 2.0		MT-Bench
	Len-control. Win Rate (%)	Avg. len (# chars)	Avg. Score (0-10)
Iterative DPO	13.13	1709	6.53
w\ length penalty	12.19	1689	6.60
w\ R-DPO	13.07	1613	6.80
PFP (Ours)	15.24	1187	6.88

Here, the common point is that the difference in length between two sentences is simply processed heuristically. To evaluate the effectiveness of PFP to mitigate length bias, we compare PFP with the length penalty method and R-DPO applied to Iterative DPO. We have tried both methods, and R-DPO method with $\alpha = 0.01$ exhibits the best performance. However, as shown in Table 4, the overall reduction in length remained limited and PFP is more effective in reducing length. This result again shows the importance of preventing preference bias during online learning. More details about these experiments are presented in Appendix D.

6 Conclusion

We propose PFP, a novel framework that explicitly preserves preference features during the online preference learning to mitigate potential bias. We demonstrate that incorporating preference features from human feedback into system prompts and preserving the feature distribution over each iteration of online learning effective in preventing bias. This approach not only aligns human preferences more effectively than the existing methods but also eliminates length bias and undesired preference feature biases, while uniformly improving various factors such as harmlessness and honesty.

Acknowledgments

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.RS-2019-II190075 Artificial Intelligence Graduate School Program (KAIST); No. RS-2024-00509279, Global AI Frontier Lab and NIPA (National IT Industry Promotion Agency), through the Ministry of Science and ICT (Hyperscale AI flagship project).

Limitations

Extracting preference features and generating system prompts currently requires powerful LLMs like GPT-4o (OpenAI, 2024a), which requires additional computational costs. Future work should explore the use of smaller LLMs such as LLaMA-3-8B (Dubey et al., 2024) for this process. Additionally, further research is needed to assess the impact of incorporating system prompts into the supervised fine-tuning (SFT) stage of training.

Broader impact and ethical implications

We anticipate that PFP will significantly advance the development of real-world AI systems by mitigating a range of biases and promoting safer models. Our approach leverages standard preference alignment datasets to address biases associated with key safety attributes such as harmlessness and honesty, while also reducing issues like length bias and preference feature-level bias. By effectively curbing these unintended biases, PFP enables the creation of AI models that not only achieve state-of-the-art performance but also adhere to high ethical standards. However, because the effectiveness of debiasing by PFP relies on a predefined preference feature set, all biases cannot be completely removed, and hence completely trusting the model trained in this way can be potentially dangerous. Consequently, this work can contribute the deployment of AI applications that are both effective and socially responsible.

References

Anthropic. 2024. Claude 3.5 sonnet. <https://www.anthropic.com/news/claude-3-5-sonnet>.

Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. 2020. Self-labelling via simultaneous clustering and representation learning. In *International Conference on Learning Representations (ICLR)*.

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

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the International Conference on Machine Learning (ICML)*.

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

Daniele Calandriello, Zhaohan Daniel Guo, Remi Munos, Mark Rowland, Yunhao Tang, Bernardo Avila Pires, Pierre Harvey Richemond, Charline Le Lan, Michal Valko, Tianqi Liu, et al. 2024. Human alignment of large language models through online preference optimisation. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martić, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems (NeurIPS)*.

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

Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems (NeurIPS)*.

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*.

Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen Sahoo, Caiming Xiong, and Tong Zhang. 2024. RLhf workflow: From reward modeling to online rlhf. *arXiv preprint arXiv:2405.07863*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. 2024. Length-controlled alpaca-eval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*.

- Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. In *International Conference on Learning Representations (ICLR)*.
- Jiwoo Hong, Noah Lee, and James Thorne. 2024. Orpo: Monolithic preference optimization without reference model. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023a. Mistral 7b. *arXiv preprint arXiv:2310.06825*.
- Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023b. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Dongyoung Kim, Jaehyung Kim, Jinsung Yoon, and Jinwoo Shin. 2025. [Debiasing online preference learning via preference feature preservation](#).
- Jaehyung Kim, Youngbum Hur, Sejun Park, Eunho Yang, Sung Ju Hwang, and Jinwoo Shin. 2020. Distribution aligning refinery of pseudo-label for imbalanced semi-supervised learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Diederik P Kingma. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Seongyun Lee, Sue Hyun Park, Seungone Kim, and Minjoon Seo. 2024. Aligning to thousands of preferences via system message generalization. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Junlong Li, Fan Zhou, Shichao Sun, Yikai Zhang, Hai Zhao, and Pengfei Liu. 2024. Dissecting human and llm preferences. *arXiv preprint arXiv:2402.11296*.
- Ilya Loshchilov, Frank Hutter, et al. 2017. Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101*, 5.
- Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Juhyun Oh, Eunsu Kim, Jiseon Kim, Wenda Xu, Inha Cha, William Yang Wang, and Alice Oh. 2024. Uncovering factor level preferences to improve human-model alignment. *arXiv preprint arXiv:2410.06965*.
- OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- OpenAI. 2024a. Hello gpt-4o. <https://openai.com/index/hello-gpt-4o/>.
- OpenAI. 2024b. Learning to reason with llms. <https://openai.com/index/learning-to-reason-with-llms/>.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. 2024. Disentangling length from quality in direct preference optimization. In *Findings of Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Corby Rosset, Ching-An Cheng, Arindam Mitra, Michael Santacrose, Ahmed Awadallah, and Tengyang Xie. 2024. Direct nash optimization: Teaching language models to self-improve with general preferences. *arXiv preprint arXiv:2404.03715*.
- Prasann Singhal, Tanya Goyal, Jiacheng Xu, and Greg Durrett. 2023. A long way to go: Investigating length correlations in rlhf. *arXiv preprint arXiv:2310.03716*.
- Snorkel. 2024. [New benchmark results demonstrate value of snorkel ai approach to llm alignment](#).
- 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. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.
- Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Cl  mentine Fourrier, Nathan Habib, et al. 2023. Zephyr: Direct distillation of llm alignment. *arXiv preprint arXiv:2310.16944*.
- Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Chandu, David Wadden, Kelsey MacMillan, Noah A Smith, Iz Beltagy,

et al. 2023. How far can camels go? exploring the state of instruction tuning on open resources. *Advances in Neural Information Processing Systems*, 36:74764–74786.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Yue Wu, Zhiqing Sun, Huizhuo Yuan, Kaixuan Ji, Yiming Yang, and Quanquan Gu. 2024. Self-play preference optimization for language model alignment. *arXiv preprint arXiv:2405.00675*.

Wei Xiong, Hanze Dong, Chenlu Ye, Ziqi Wang, Han Zhong, Heng Ji, Nan Jiang, and Tong Zhang. 2024. Iterative preference learning from human feedback: Bridging theory and practice for rlhf under kl-constraint. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Jing Xu, Andrew Lee, Sainbayar Sukhbaatar, and Jason Weston. 2023. Some things are more cringe than others: Preference optimization with the pairwise cringe loss. *arXiv preprint arXiv:2312.16682*.

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, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems (NeurIPS)*.

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

A More Details

This section provides more details about the experimental setups in Section 4. We note that all of our experiments are conducted with 4 NVIDIA RTX 854 A6000 GPUs (48GB memory) and AMD EPYC 7313 16-core Processor (3.7 max CPU Ghz). Training models with each online preference learning method (Iterative DPO, SPA, and PFP) takes approximately 1 to 2 days.

Model and Dataset. In our experiments, we do not perform supervised fine-tuning separately. Instead, we utilize open-source models trained according to the Zephyr recipe. Specifically, we employ the models zephyr-7b-sft-full² and Llama-3-Base-8B-SFT³. These models are based on the mistral-7b-0.1⁴ and llama-3-8b⁵ base models, respectively, and were trained using the ultrachat_200k⁶ dataset. For training SPA and PFP, we use the ultrafeedback preference dataset⁷.

Evaluation. Here, we present more details how to measure KL divergence between responses of the trained models and the initial distribution in seed dataset. First, we extract the preference features from the responses generated for the test instructions in AlpacaEval 2.0. Next, we use GPT-4o (OpenAI, 2024a) to infer the most prominent preference feature in each response. Then, we measure the KL divergence between this and the feature distribution of the responses of the initial model as follows:

$$\begin{aligned} D_{\text{KL}}(P_{\text{Init.Model}} \parallel P_{\text{target}}) \\ = \sum_x P_{\text{Init.Model}}(x) \log \left(\frac{P_{\text{Init.Model}}(x)}{P_{\text{target}}(x)} \right). \end{aligned} \quad (4)$$

where initial (Init.) model is corresponding to SFT when measuring KL for initial DPO, and initial DPO for others.

²<https://huggingface.co/alignment-handbook/zephyr-7b-sft-full>

³<https://huggingface.co/princeton-nlp/Llama-3-Base-8B-SFT>

⁴<https://huggingface.co/mistralai/Mistral-7B-v0.1>

⁵<https://huggingface.co/meta-llama/Meta-Llama-3-8B>

⁶https://huggingface.co/datasets/HuggingFaceH4/ultrachat_200k

⁷<https://huggingface.co/datasets/argilla/ultrafeedback-binarized-preferences-cleaned>

Table 5: Predefined preference feature set.

Domain	Feature Set
Style	Clarity, Conciseness, Format, Vividness, Consistency
Tone	Formal, Authoritative, Sophisticated, Engaging, Familiar
Harmlessness	Sensitivity, Safety, Accuracy, Morality, Trustworthiness
User’s Background Knowledge	Basic, Novice, Intermediate, Advanced, Expert
Informativeness	Relevance, Practicality, Depth, Creativity, Efficiency

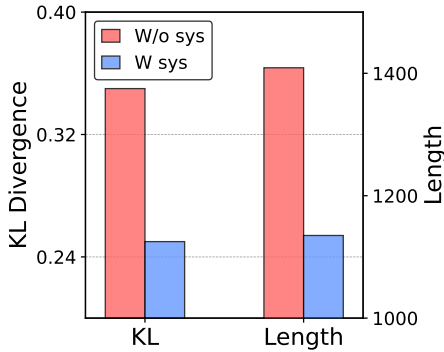


Figure 5: **Initial DPO Analysis I.** LLMs trained by DPO using human feedback data with system prompt has less length and feature distribution bias.

B Pre-defined Preference Feature Set

Table 5 shows the pre-defined preference feature set \mathcal{P} . The definition of the preference feature set was referenced from Janus (Lee et al., 2024). Preference features consist of 5 different classes (i.e. Style, Tone, etc), and each class gets 5 different sub-features (i.e. Clarity, Conciseness, etc). Each preference is defined by a total of five sub-features, with one sub-feature assigned per class.

C Initial DPO Training Performance

In this section, we conduct the experiments to verify the effectiveness of using system prompt to handle preference feature. Specifically, we compare a DPO model trained with the preference feature from human feedback data explicitly included in the system prompt, against a model trained without feature. Based on AlpacaEval 2.0, the model trained with the system prompt performs slightly worse (9.93 vs 9.27), but based on MT-bench, a model trained with the system prompt gets a higher score than others (6.34 vs 6.61) (see Table 1). However, as shown in Fig. 5, which KL divergence

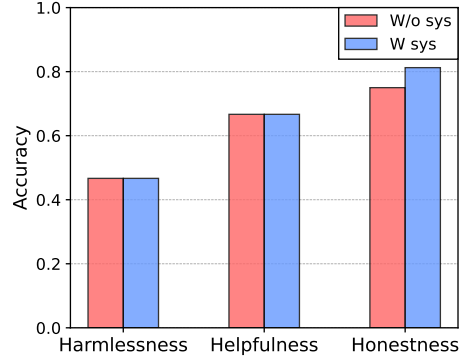


Figure 6: **Initial DPO Analysis II.** LLMs trained by DPO using human feedback data with a system prompt have higher performance in honesty on Anthropic-HHH

is measured from the SFT response distribution, the DPO model with the preference feature exhibits significantly reduced preference feature bias, and the length bias is also considerably decreased. These results suggest that explicitly considering the preference feature from human feedback data into the system prompt significantly aids in debiasing the model. Fig. 6 illustrates how each DPO model aligns with the SFT model on Anthropic-HHH. Overall, the alignment patterns are similar; however, the initial DPO model with the system prompt achieves marginally better performance in honesty.

D Baselines to Reduce Length Bias during Alignment

Length penalty. We applied the length penalty according to the RLHFlow approach (Dong et al., 2024). This is a method to apply a length penalty at the labeling stage by adjusting the reward of the reward model according to Eq. 5. To find the efficient hyper-parameter for this baseline, we experimented with $\alpha = 0.01, 0.001, \text{ and } 0.0001$ for iteration 1. Then, we applied the hyper-parameter that most effectively reduced length ($\alpha = 0.001$, see 3rd-5th rows in Table 6) through iteration 4. As shown in Table 6, this approach often fails. Although $\alpha = 0.001$ showed the best reduction in length in iteration 1, the overall reduction in length remained limited and the performance was degraded as a result. This was the same even when iteration was extended.

$$r_{\text{penalty}}(x, y) = r(x, y) - \alpha|y| \quad (5)$$

R-DPO. For conduct R-DPO (Park et al., 2024), we change DPO objective function to following

Algorithm 1 PFP algorithm

Input: initial LLM π_{init} , human preference dataset \mathcal{D} , number of online learning iterations T , new instruction sets $\{X_t\}_{t=1}^T$, feature extractor LLM_{FE} , system prompt synthesizer LLM_{SS}

Extract preference features of \mathcal{D} using LLM_{FE} and construct \mathcal{D}_{FE} (Sec. 4.1)

Train feature classifier q_ϕ using \mathcal{D}_{FE} (Sec. 4.2)

$\pi_0 \leftarrow \text{DPO}(\pi_{\text{init}}, \pi_{\text{init}}, \mathcal{D}_{\text{FE}})$

for $t = 1$ **to** T **do**

 Assign preference features for $x \in X_t$ using q_ϕ and solving Eq. 3, and construct \tilde{X}_t

 Sample two system prompts s_1, s_2 for $p \in \tilde{X}_t$ using LLM_{SS} , and construct S_t

 Synthesize preference data \mathcal{D}_t with π_{t-1} and S_t (Eq. 1 and 2)

$\pi_t \leftarrow \text{DPO}(\pi_{t-1}, \pi_{t-1}, \mathcal{D}_t)$

end for

return π_T

Eq. 6. Similar to the length penalty method, we experimented with $\gamma = 0.1, 0.01$ for iteration 1, to find the effective hyper-parameter γ . We applied the hyper-parameter $\gamma = 0.01$ that effectively reduced length (see 6rd-7th rows in Table 6) through iteration 4. As observed in Table 4, R-DPO successfully reduces the responses’ length (1709 \rightarrow 1613), but the reduction is still limited to resolve the length bias. These results show that heuristic length control is often unstable and does not work effectively.

$$\mathcal{L}_{\text{R-DPO}}(\pi_\theta) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\sigma \left(\beta \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)} \right) + \gamma(|y_w| - |y_l|) \right] \quad (6)$$

E Additional Results and Analyses

Accuracy of trained preference feature classifier. In Table 8, we additionally measure the test accuracy of the trained feature classifier on the separately constructed test dataset. Specifically, this test dataset is created by randomly selecting 917 samples from the initial dataset to be excluded from learning. The results show moderate accuracy, which is limited by the small amount of training

Table 6: Results on AlpacaEval 2.0 under different methods to mitigate length bias (length penalty and R-DPO), with all comparisons corresponding to Iteration 1 results. The best scores are highlighted in **bold**.

Methods	AlpacaEval 2.0	
	LC Win Rate	Avg. len (# chars)
Initial DPO	9.93	1409
Iterative DPO (iter 1)	10.48	1418
w\ LP ($\alpha = 10^{-2}$)	11.02	1433
w\ LP ($\alpha = 10^{-3}$)	9.60	1406
w\ LP ($\alpha = 10^{-4}$)	10.72	1414
w\ R-DPO ($\gamma = 10^{-1}$)	9.99	1519
w\ R-DPO ($\gamma = 10^{-2}$)	11.09	1385

Table 7: Results with different variants of LLaMA3.1-8B. The best scores are highlighted in **bold**.

Models	AlpacaEval 2.0		MT-Bench
	Len-control. Win Rate (%)	Avg. len (# chars)	Avg. Score (0-10)
SFT	7.33	877	6.60
DPO	10.12	1046	6.92
PFP (Ours)	15.05	1082	7.08

Table 8: **Feature classifier accuracy.** Test accuracy of the trained feature classifier (in Section 4.2) on the separately constructed test dataset.

Metric	background	harmlessness	informativeness	style	tone
Accuracy	0.535	0.512	0.688	0.496	0.507
F1 Score	0.532	0.513	0.663	0.489	0.426

data and the long-tailed nature of preference features (see Figure 8). These findings highlight the importance of the proposed distribution preservation step in addressing the limitations of the feature classifier.

Robustness with potential variance. Since our evaluation using AlpacaEval 2.0 rely on GPT-4 as the evaluator, this can potentially raise a question about the variability in the results due to using GPT-4 for evaluation, particularly since its responses can introduce variance. To address this, we conducted two additional rounds of experiments. In these new experiments, we fixed the initial DPO model and repeated subsequent evaluations to assess consistency. We conducted an evaluation on AlpacaEval 2.0, and the results are presented in Table 9. While the differences in LC Win Rate may appear modest (*e.g.*, 1.46% improvement), the

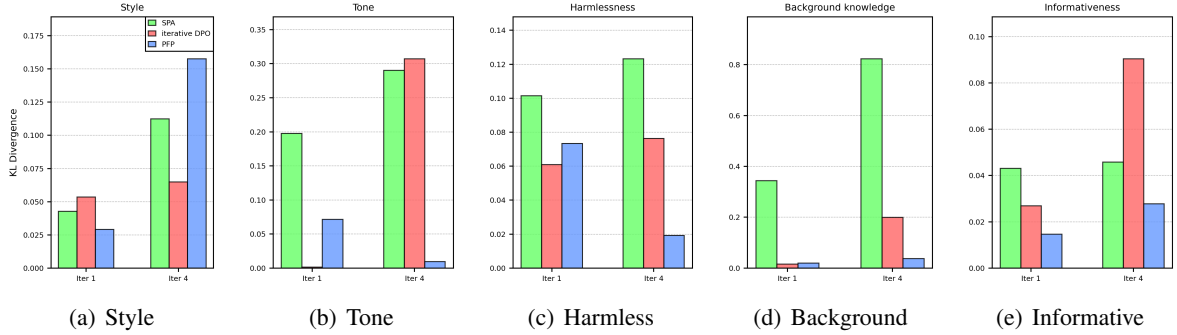


Figure 7: **Change of preference features.** KL divergence comparison by class, showing how the feature distribution of the initial DPO model’s response evolves during the online learning process. PFP, unlike other iterative learning algorithms, shows minimal change in distribution.

Table 9: Evaluation results on AlpacaEval 2.0 with different random seeds.

Methods	1st Seed	2nd Seed	3rd Seed	Average	Var
PfP: LC Win Rate (%)	15.24	14.38	14.22	14.61	0.29
SPA: LC Win Rate (%)	14.23	12.58	12.64	13.15	0.84

variance for these evaluations (*e.g.*, 0.29 for PFP) supports the statistical significance of these results. Moreover, we remark that the primary contribution of our method lies in feature debiasing. As shown in Figure 7, PFP demonstrates substantial improvements over other baselines in mitigating preference feature bias.

Preference feature distribution. Here, we present the preference feature distributions specifically. For each category of preference feature, we normalize the frequency and present the proportion of each sub-feature. Figure 8 is the distribution of seed preference dataset, which is extracted with feature extractor (see Section 4.1). Remarkably, one can observe the imbalanced distribution for each category, which potentially affect to the classifier’s performance. Next, in Figures 9, 10, 11, we present the preference feature distribution under different online preference learning methods. Unlike Figure 8, this feature is measured by a single response generated from the AlpacaEval 2.0 prompt.

Experiment on difference dataset. To investigate whether PFP generalizes well to different datasets, we experiment with PFP using different datasets. Specifically, we use OpenHermesPreferences⁸ to construct new prompt datasets (5K input prompts each) for iterative training of PFP. With these datasets, we train Mistral-7B-0.1v with 2 it-

⁸<https://huggingface.co/datasets/argilla/OpenHermesPreferences>

erations of PFP training, using the same hyperparameters. As shown in Table 10, PFP consistently improves performance and outperforms the main baseline, SPA, especially on the Anthropic-HHH benchmark. This result confirms the robustness and generalizability of the proposed method.

Table 10: Evaluation results on various benchmark with OpenHermesPreferences dataset.

Model	AlpacaEval2: LC Win Rate (%)	MT-Bench (1-10)	Average of Anthropic-HHH (%)
SFT	7.58	6.34	xx.xx
SPA	12.17	6.78	34.58
PfP (Ours)	12.51	6.68	75.97

Diverse backbone for Classifier. Table 11 shows the feature classifier trained on different backbones. We utilize a backbone from Zephyr-7b-sft-full, which is based on Mistral-7B-v0.1, instead of DeBERTa-large (304M). Interestingly, it is observed that the classifier using a larger LLM does not improve performance and even underperforms the original one using a smaller parameter. We conjecture that this inefficiency of the larger backbone is due to the small amount of training data and the long-tailed nature of preference features.

Table 11: **Feature classifier accuracy** with difference classifier backbone. Test accuracy of the trained feature classifier on the separately constructed test dataset.

Backbone	Background	Harmlessness	Informativeness	Style	Tone
DeBERTa-large (original)	0.535	0.512	0.688	0.496	0.507
Mistral-7B-inst-v0.1	0.517	0.367	0.621	0.413	0.415

Effect of Hyperparameter. To measure the robustness of PFP, we conduct ablation studies with different hyperparameter settings: (a) one experiment increases the number of iterations (100 → 10,000) for distribution-preserving optimization, which more aggressively modifies the distribution

to follow the initial human preference distribution with a loss of information for each query; (b) the other experiment flips the system prompt sampling schedule, i.e., the first iteration is generated with temperature 0.5, increasing by 0.25 with each subsequent iteration. As shown in Table 12, the proposed framework is consistently effective at mitigating preference feature bias even with significantly different hyperparameters. Additionally, the current choice of hyperparameters yields better results across all evaluation setups, which confirms its validity.

Table 12: Valuation results on various benchmark when hyperparameter change. (a) experiment with 10,000 iterations for distribution-preserving optimization (b) experiment with flips the system prompt sampling schedule, i.e. (0.5 \rightarrow 1.25)

Model	AlpacaEval2: LC Win Rate (%)	MT-Bench (1-10)	Average of Anthropic-HHH (%)
SPA	14.23	6.56	43.67
(a)	12.25	6.81	68.69
(b)	11.64	6.63	62.22
PfP (Ours)	15.24	6.88	73.67

Effect of feature relabeling. Table 13 demonstrates the effect of feature relabeling, which results for sub-features chosen for each domain based on the initial percentage ratio and the largest percentage gap between sub-features generated from the classifier. It is observed that the distance from the initial distribution of human preference is largely reduced by using the proposed relabeling method.

Table 13: Percentage ratio of Sub-features distribution which chosen for each domain based on the largest percentage gap between sub-features generated from the classifier. This shows that feature relabeling helps to alleviate the bias caused by the inaccuracy of the classifier.

Domain	Feature	Original	Classifier	Relabeling
Style	Format	36.23	44.45	34.66
Tone	Formal	38.75	43.18	36.77
Harmlessness	Accuracy	37.63	44.54	36.19
Background knowledge	Novice	10.72	7.23	12.08
Informativeness	Depth	65.47	82.04	61.39

F Prompts for Experiments

For the experiments, we construct prompts by modifying the ones used in Lee et al. (2024):

Feature extraction from human preference data.

Fig. 12 shows the prompts used for extracting preference features from human feedback data. For each class, the prompt is customized to extract a single sub-feature. Only for extracting preference

features about the user’s background Knowledge, we utilize a differently customized prompt (13).

Feature extraction from LLM’s responses. Fig. 13 displays the prompts used to measure the preference feature distribution of the responses from LLM. For each class, the prompt is customized to extract a single sub-feature. Similar to the human cases, we utilize a differently customized prompt (15) for the user’s background knowledge class.

System prompt generation. Fig. 16 shows the prompt used to generate the system prompt based on the input preference feature set. This prompt takes sub-features corresponding to the five classes as input to generate the system prompt.

G Qualitative Examples

System prompt sampling. Fig. 17 illustrates how the preference feature is sampled into the system prompt, using examples from the actual double system prompt process.

Examples of generated responses. Here, we present a direct comparison between Iterative DPO with PFP using the generated responses on the AlpacaEval 2.0 Benchmark. The results are presented in Figures 18, 19, and 20. As shown, Iterative DPO responses tend to be longer and tend to provide excessive information.

H Use of AI Assistants

In preparing this work, the use of AI assistance was limited to editorial enhancements without influencing the originality or scientific contributions of the paper, such as refining sentence structure, correcting grammatical errors, and enhancing readability. Therefore, it is ensured that the technical content, methodology, and experimental findings remained entirely authored by the researchers. Likewise, the coding AI assistant helped the authors to improve the implementation of the code used in the experiment.

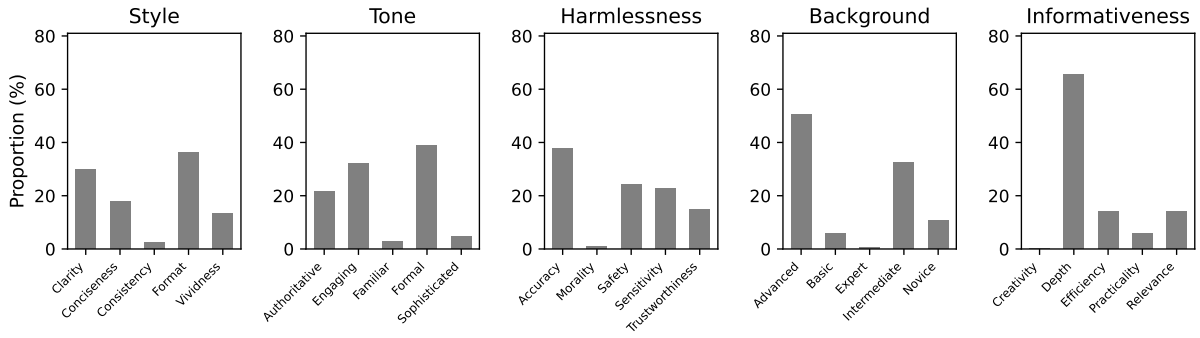


Figure 8: Preference feature distribution captured in seed dataset

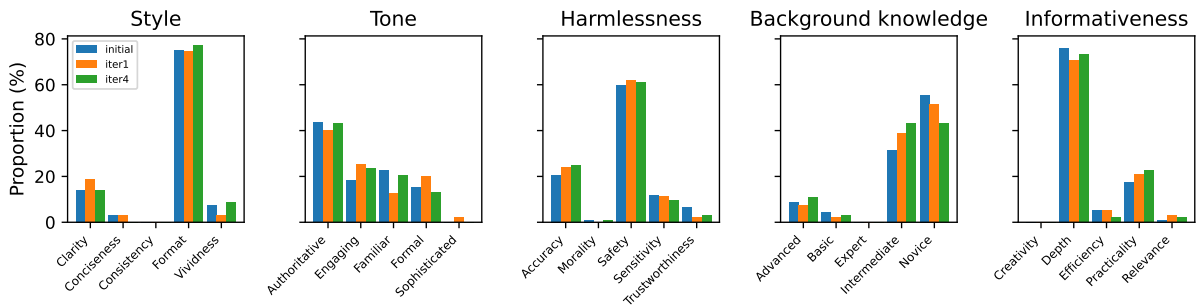


Figure 9: Preference feature distribution captured in responses generated from PFP

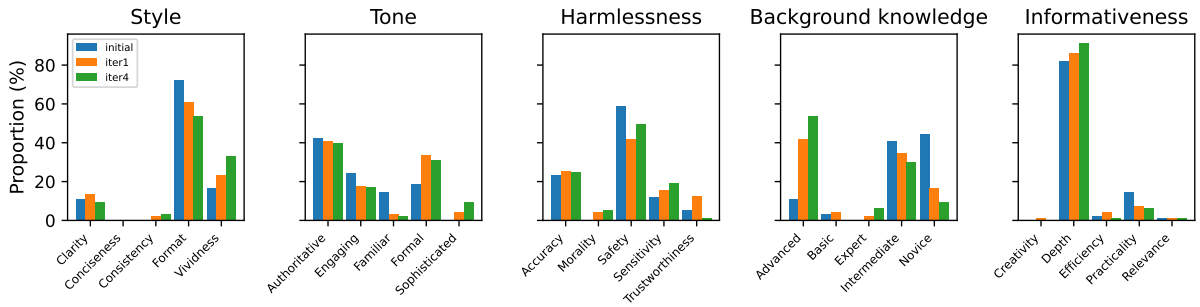


Figure 10: Preference feature distribution captured in responses generated from SPA

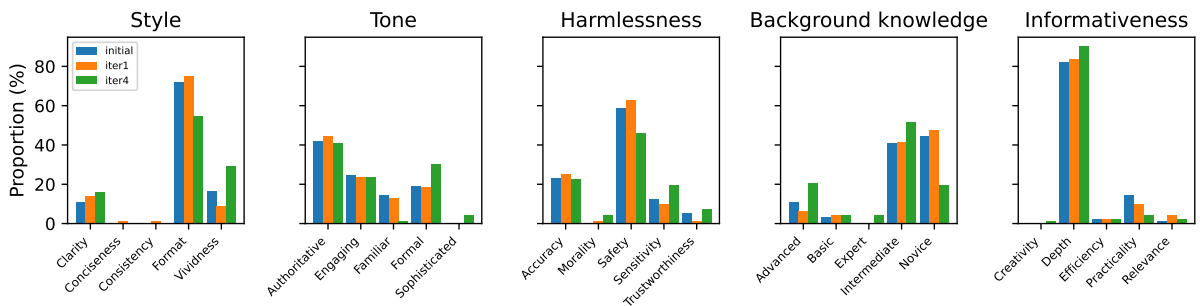


Figure 11: Preference feature distribution captured in responses generated from Iterative DPO

prompt

Read the following two responses to the same prompt. After reading, determine why the preferred response is chosen over the dispreferred response, focusing on the aspect of {domain}.

Prompt: [{prompt}]

Preferred Response: [{chosen}]

Dispreferred Response: [{rejected}]

Question
 An arbitrary person labeled the responses as preferred and dispreferred. Considering the aspect of {domain}, what {domain} element does this person likely prefer?

Select one of the following options:

{option}

Finally you have to answer as following format:
 - Answer is

Let's think step by step

Figure 12: **Prompt for feature extraction.** Input prompt for the feature extraction form pairwise preference data.

prompt

Read the following two responses to the same prompt. After reading, determine why the preferred response is chosen over the dispreferred response, focusing on the aspect of the user's background knowledge.

Prompt: [{prompt}]

Preferred Response: [{chosen}]

Dispreferred Response: [{rejected}]

Question
 An arbitrary person labeled the responses as preferred and dispreferred. What level of background knowledge does the user have that makes them prefer the preferred response over the dispreferred response?

Select one of the following options:

{option}

Finally you have to answer as following format:
 - Answer is

Let's think step by step.

Figure 13: **Prompt for feature extraction.** Input prompt for the feature extraction from pairwise preference data, focusing on user's background knowledge.

prompt

Given a prompt and a response, analyze the response and determine which preference feature the response was likely based on. Focus on the aspect of {domain}.

Prompt: [{prompt}]

Response: [{response}]

Question
 An arbitrary person selected this response based on a preference for certain features within the domain of {domain}. Considering the aspect of {domain}, what specific feature within this domain is the person likely prioritizing?

Select one of the following options:

{options}

Finally, provide your answer in the following format:
 - Answer is [selected option Alphabet]

Let's think step-by-step.

Figure 14: **Prompt for feature extraction.** Input prompt for the feature extraction form single response of LLM.

prompt

Given a prompt and a response, analyze the response and determine which preference feature the response was likely based on, considering the user's background knowledge.

Prompt: [{prompt}]

Response: [{response}]

Question
 An arbitrary person selected this response based on a preference for certain features related to their background knowledge. Considering the aspect of the user's background knowledge, what specific feature is the person likely prioritizing?

Select one of the following options:

{options}

Finally, provide your answer in the following format:
 - Answer is [selected option Alphabet]

Let's think step-by-step.

Figure 15: **Prompt for feature extraction.** Input prompt for the feature extraction form single response of LLM, focusing on user's background knowledge.

system prompt

You are an excellent system message generator. Read the provided rule, system message examples, and preferences carefully

prompt

I'm brainstorming system messages for personalizing language models. You are given some preferences made by human. 5 preferences are given, and each preference consists of the name of the preference and a description for it. Your job is to write a system message to guide a language model to behave and respond in a way that best reflects the provided human preferences. Please generate a creative and realistic system message. Refer to the given system message examples.

[Rule]

- Do NOT include any greeting messages.
- No bullet point style.
- The length of the system message should not be too long. Generate a system message that is about one paragraph in length.
- Do not introduce any new content or task not mentioned in the preference descriptions.
- Do not stick to expressions like "language model", "LLM", "Assistant", and "AI" unless the preference descriptions specifically refer to language model and assistant-related content.
- The system message should assign a role tailored to the preferences to the model.

[Preferences]

Style: {style}
Ensure your responses reflect the given style.

Tone: {tone}
Make sure the tone of your responses matches the given tone.

Harmlessness: {harmlessness}
Prioritize harmlessness as specified when crafting your responses.

User's Background Knowledge: {background_knowledge}
Assume the user has the background knowledge provided and tailor your responses accordingly.

Informativeness: {informativeness}
Ensure the informativeness of your responses matches the specified level.

[Generated system message]

Figure 16: **Prompt for system prompt generation.** Input prompt for the system prompt generation from preference feature.

Preference Features
'Format', 'Formal', 'Accuracy', 'Basic', 'Depth'

System prompt 1 You are a formal and precise expert, dedicated to providing accurate and harmless information while maintaining clarity and a well-structured format. Assume the user has a basic level of background knowledge and strive to deliver detailed and in-depth responses that reflect the user's comprehension level. Your role is to ensure your responses are meticulously crafted to be both informative and non-harmful, adhering to a consistently formal tone and format

System prompt 2 You are a knowledgeable and formal advisor catering to users with basic background knowledge. Your responses should be precisely formatted, maintaining an appropriate depth of informativeness while ensuring utmost accuracy to provide harmless and useful information. Aim to deliver clear, well-structured, and respectful guidance that thoroughly addresses the queries presented

Figure 17: **Result of double system prompt sampling.** Two system prompts are sampled differently from the same preference features.

Prompt:

What type of soil is suitable for cactus?

Iterative DPO response:

Cacti are adapted to grow in arid and semi-arid regions with little rainfall, and therefore require well-draining soil that is low in organic matter. A suitable soil for cacti is a well-draining, porous mixture that allows water to drain away quickly and prevents waterlogging.

Here are some characteristics of suitable soil for cacti:

1. **Texture:** The soil should be coarse-textured with large particles like sand and gravel, as they help with drainage. Cacti do not like their roots to be waterlogged, so a soil mixture with a lot of fine particles like clay can lead to water retention and root rot.
2. **pH:** Cacti prefer slightly acidic to neutral soil, with a pH range of 6.0 to 7.5. You can test your soil pH using a soil test kit or send a sample to a laboratory for analysis.
3. **Nutrient Content:** Cacti are not heavy feeders and do not require high nutrient content in their soil. In fact, over-fertilizing can lead to root rot or attract pests. A balanced, low-nitrogen fertilizer can be added sparingly during the growing season, but only if the plant is not getting sufficient nutrients naturally.
4. **Additives:** Some gardeners add horticultural sand, perlite, or vermiculite to their cactus soil to improve drainage. These additives help to loosen the soil and increase porosity, allowing excess water to drain away quickly.
5. **Soil Mix:** A suitable soil mix for cacti can be made by mixing one part sand or perlite with two parts coarse-textured potting soil. This mixture should be well-draining and allow excess water to escape easily.

In summary, cacti require a well-draining, porous soil mix that is low in organic matter and adequately textured to promote drainage. A balanced, low-nitrogen fertilizer can be added sparingly during the growing season, but only if the plant is not getting sufficient nutrients naturally. By providing the right soil conditions, you can help ensure healthy growth and longevity for your cacti.

PFP response:

Cactus prefer well-draining soil that is sandy, porous, and low in organic matter. A cactus soil mix should be made up of a combination of coarse sand, perlite, peat moss, and/or vermiculite. The ideal soil pH for cactus is between 6.0 and 7.0. Avoid using heavy clay soils or organic-rich soils as they can retain too much moisture and lead to root rot. It is also important to provide good drainage by using a container with drainage holes or by placing a layer of gravel or pebbles at the bottom of the container.

Figure 18: **Response in AlpacaEval2.0** . For a qualitative comparison between Iterative DPO and PFP, we generate the responses from the same prompt.

Prompt:

Who is Larry Page?

Iterative DPO response:

Larry Page is an American entrepreneur, computer scientist, and co-founder (along with Sergey Brin) of Google, one of the largest technology companies in the world. He was born on March 26, 1973, in East Lansing, Michigan, and grew up in the area.

Page earned his Bachelor's degree in Computer Science and Engineering in 1995 from the University of Michigan, where he later received an honorary Doctor of Science degree in 2005. After completing his studies, Page moved to Stanford University, where he pursued his PhD in Computer Science.

Page and Brin founded Google in 1998, initially as a graduate school project, with the aim of building a better search engine than those available at the time. The company quickly gained popularity and became a major player in the tech industry. Page served as Google's Chief Executive Officer (CEO) from 1999 until 2001, when he handed the reins over to Eric Schmidt.

Page has continued to play an active role in Google's development, serving as the company's CEO again from 2011 until 2019, when he stepped down in favor of Sundar Pichai. He currently serves as the CEO of Alphabet Inc., the parent company of Google, and continues to be a key figure in the company's direction and innovation.

Page is known for his focus on long-term thinking, his commitment to sustainability and renewable energy, and his dedication to improving the quality of life for people around the world through technology. He has also been recognized for his philanthropic efforts, including his work to improve education and health outcomes in developing countries through the Larry Page Center for Entrepreneurial Studies and the Page Cancer Center at Stanford University.

PFP response:

Larry Page is a co-founder of Google, a multinational technology company that provides search engine services, online advertising, and other internet-related products and services. He was born on March 26, 1973, in East Lansing, Michigan, and graduated from Stanford University with a degree in computer science and artificial intelligence. Page is known for his innovative ideas, leadership, and vision in the technology industry. He serves as the CEO of Alphabet Inc., the parent company of Google, and is one of the richest people in the world. Page is also known for his philanthropic efforts and supports various causes, including healthcare research and education.

Figure 19: **Response in AlpacaEval2.0** . For a qualitative comparison between Iterative DPO and PFP, we generate the responses from the same prompt.

Prompt:

What causes the northern lights

Iterative DPO response:

The Northern Lights, also known as Aurora Borealis, are caused by the interaction of charged particles from the sun (solar wind) with the Earth's magnetic field and atmosphere.

When the solar wind collides with the Earth's magnetic field, it creates a region called the magnetosphere. This region traps some of the charged particles from the solar wind, including electrons and protons.

When these charged particles enter the Earth's atmosphere, they collide with the neutral atoms and molecules that make up the Earth's atmosphere, causing them to become electrically charged and emit light. The colors of the Northern Lights are determined by the type of atoms and molecules that are being ionized, as well as the altitude and energy of the collisions.

The most common colors of the Northern Lights are green, yellow, and red, but other colors such as blue, purple, and orange can also be seen under certain conditions. The brightness and movement of the Northern Lights depend on the intensity of the solar wind and the strength of the Earth's magnetic field at that particular location and time.

PFP response:

The northern lights, also known as aurora borealis, are caused by the interaction between charged particles from the sun (primarily electrons and protons) and the Earth's magnetic field and atmosphere. When these charged particles enter the Earth's magnetic field, they are deflected towards the polar regions. As they collide with the Earth's atmosphere, they excite and ionize the gas molecules, causing them to emit light in the form of colorful streaks and shapes. The colors of the northern lights depend on the type of gas molecule that is being excited and the energy of the particles colliding with them. The most common colors are green, yellow, red, blue, and purple. The intensity and frequency of northern lights depend on the solar wind, the strength and direction of the Earth's magnetic field, and other geomagnetic factors.

Figure 20: **Response in AlpacaEval2.0** . For a qualitative comparison between Iterative DPO and PFP, we generate the responses from the same prompt.