Probability Flow Optimization: Mitigating Squeezing Effect for Efficient Binary Preference Alignment

Anonymous ACL submission

Abstract

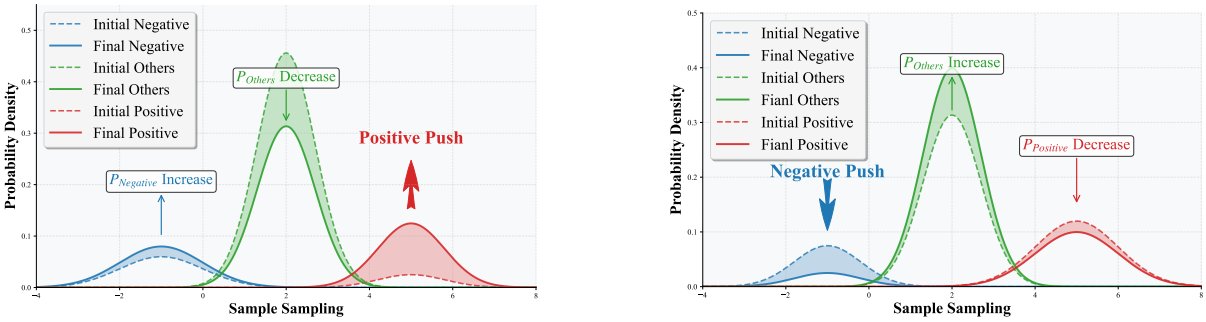
Unlike pairwise optimization, which jointly processes sample pairs, binary preference alignment optimizes positive and negative samples independently. This fundamental limitation introduces two critical issues. First, penalizing negative samples provides no guarantee that the released probability mass will flow toward positive samples; instead, it may inadvertently reduce the likelihood assigned to positive samples. Second, the one-sided labels in binary preference alignment lack contrastive signals, rendering negative samples ambiguous and potentially insufficient to warrant rejection. Training on such ambiguous negatives exacerbates the suboptimality of probability flow. To address these issues, we track and analyze the evolution of probability quality throughout binary preference alignment training, and propose a novel probability flow optimization perspective that mitigates the aforementioned adverse probability dynamics. Extensive experiments on dialogue benchmarks including MT-Bench and AlpacaEval 2, as well as on recommendation datasets where unclicked items are not necessarily absolute negatives, demonstrate that our method consistently outperforms existing binary preference alignment approaches.

1 Introduction

Using preference data to fine-tune large language models (LLMs) is crucial to align them with human preferences and make them more helpful and less harmful (Bai et al., 2022). In recent years, preference alignment training for LLMs has been extensively explored. The researchers trained a reward model on human preference data to evaluate the responses of LLMs and first proposed the Reinforcement Learning from Human Feedback (RLHF) paradigm to align large language models with human preferences (Ouyang et al., 2022). Subsequently, the research community is steadily expanding its focus on employing reinforcement learning algorithms to train an LLM policy that maximizes the quality score of its responses to specific prompts, thus aligning the generated content with human preferences (Yue et al., 2025; Yu et al., 2025a; Yan et al., 2025).

Nevertheless, resource constraints often restrict many real-world scenarios to offline preference data only. Approaches such as DPO (Rafailov et al., 2023), IPO (Azar et al., 2024), and SIMPO (Meng et al., 2024) leverage human-annotated offline preference data and replace reinforcement learning with supervised learning. Although these methods significantly reduce the resource requirements for preference training, they still depend on pairwise labeled data. Given that online feedback is typically tied to displayed responses, developing methods that can utilize low-cost single-label training data to achieve better alignment with human preferences is essential. Motivated by this, KTO (Ethayarajh et al., 2024) introduces a binary preference loss that learns from single-label data without requiring pairwise comparisons.

Although the KTO algorithm supports binary preference alignment, its inherent binary labeling scheme restricts each training iteration to a unidirectional update, i.e., either increasing the probability of the positive sample or decreasing that of the negative sample. Consistent with the observations of Ren et al. (Ren and Sutherland, 2024), we find in our experiments that the flow of probability mass may deviate from the expected trajectory during binary preference alignment. As illustrated in Figure 1, when the negative sample already has a very low probability during optimization, the probability mass released from it may not be effectively transferred to the positive sample; instead, it may flow to other non-target samples, resulting in an insufficient increase in the probability of positive samples or even a possible decrease. Conversely, when the positive sample has a low probability, the absorbed probability mass tends to come predom-



(a) The squeezing effect of positive training.

(b) The squeezing effect of negative training.

Figure 1: The squeezing effect when training with extremely low probability samples. When training on samples with low probability, the squeezing effect may lead to insufficient encouragement for positive samples and insufficient punishment for negative samples.

inantly from non-target samples rather than from effectively reducing the negative sample's probability, thereby weakening the suppression of the negative sample. Furthermore, since binary labels lack a comparative reference, negative samples are not necessarily absolutely rejectable, which renders the original labels potentially ambiguous and further reduces training efficiency.

In this paper, we propose an optimization perspective for binary preference alignment based on probability flow analysis, emphasizing that mitigating the squeezing effect is essential in binary preference alignment. Building on this perspective, we propose a binary preference alignment algorithm, termed Probability Flow Optimization (PFO), to improve the efficiency of probability flow. We further provide a theoretical justification for PFO, demonstrating that our optimization objective serves as a lower bound of the Rényi $\mathcal{D}_2$ divergence. Furthermore, to address the inherent label ambiguity in certain real-world binary alignment scenarios where a negative label may not necessarily imply absolute rejection, we propose a variant of PFO termed PFOP (PFO Positive). This variant follows the principle of balancing the squeezing effect and the training effect. Compared to PFO, this variant sacrifices a portion of squeezing mitigation in favor of encouraging the training effect on positive samples. As such, PFOP is particularly well suited for scenarios such as recommendation systems, where positive feedback carries substantially greater practical significance than negative feedback.

Finally, we experimentally demonstrate that the PFO algorithm outperforms existing methods across multiple models on the MT-Bench (Zheng et al., 2023) and AlpacaEval 2 (Li et al., 2023) benchmarks. Furthermore, we validate that the PFO variant PFOP demonstrates superior preference understanding and recommendation capabilities compared to existing methods on the open-source recommendation datasets MovieLens-25M (Harper and Konstan, 2015) and Goodreads (Wan et al., 2019).

In summary, our contributions are as follows:

- We demonstrate that the squeezing effect is non-negligible in binary preference alignment. This effect causes unilateral training to adversely impact the probability of the opposite side, and we provide a detailed analysis of this phenomenon.
- We propose a novel optimization method that effectively mitigates the squeezing effect, and we prove that its essence is to optimize a second-order divergence that focuses more on high-probability regions.
- We further develop a variant that balances the squeezing effect and direct training effects to alleviate excessive penalization of negative labels, which makes it more suitable for scenarios with real-world users' subjective feedback.

2 Related Work

Reinforcement Learning from Human Feedback (RLHF) is a key technology that enhances the response quality of large language models by aligning them with human preferences. Its training process consists of three main stages. The model is first initialized by supervised fine-tuning (Zhou et al., 2023; Javaheripi et al., 2023). Then a reward model is trained based on manually labeled

preference data, to output a scalar reward signal for the generated responses of the policy (Wang et al., 2024; Liu et al., 2024; Yu et al., 2025b). Finally, reinforcement learning algorithms such as Proximal Policy Optimization (PPO) (Schulman et al., 2017) are employed to optimize the policy. In recent years, more stable alternatives have emerged, including GRPO (Shao et al., 2024), VinePPO (Kazemnejad et al., 2024) and on-policy distillation (Li et al., 2026). Despite the significant advantages that RLHF has shown in areas such as mathematical reasoning (Team et al., 2025) and safety alignment (Bai et al., 2022), its deployment remains challenging in many domains compared to traditional supervised learning due to its more complex training process and the requirement for precise rewards.

Due to the complex implementation and poor training stability of RLHF, researchers have begun to promote offline preference alignment methods based on labeled data. (Rafailov et al., 2023) proposed to implicitly define the reward function based on the likelihood of the policy, bypassing the construction of an explicit reward model. Inspired by them, a large number of offline direct preference alignment methods have been proposed, such as SIMPO (Meng et al., 2024), SIMPER (Xiao et al., 2025), and EXO (Ji et al., 2024a). Although these methods have proven effective, they typically rely on paired preference-labeled data. To address this limitation, (Ethayarajh et al., 2024) proposed the KTO algorithm, which decomposes paired preference training into binary preference training requiring only samples labeled as "good" or "bad". (Mao et al., 2024) proposed an algorithm based on fine-tuning principles to establish stable bidirectional negative feedback during optimization. (Haldar et al., 2025) introduced a preference optimization method for safety alignment from a divergence estimation perspective. However, the aforementioned approach overlooked the issue of probability flows deviating from the intended direction due to training data asymmetry, potentially leading to limitations such as insufficient positive reinforcement or inadequate negative punishment.

3 Methods

This section begins with a description of our formulation notation and relevant algorithmic background. We then discuss the probability flow issue identified in preliminary experiments. Motivated by this finding, we introduce an optimization perspective that emphasizes both the squeezing effect and its balance with the direct training effect. Under this framework, we propose two distinct approaches: one tailored to general dialogue settings, and another designed for online feedback scenarios involving negative label ambiguity.

3.1 Background

Notations. We formalize the preference dataset as $\mathcal{D} = \{(x, y_w, y_l)\}$, where each data point consists of an input prompt x and its corresponding response pair $\mathbf{y} = (y_w, y_l)$ generated by a language model, with y_w denoting the chosen response and y_l the rejected response. For binary preference alignment, each prompt x contributes exactly one response, thus inducing partitioned datasets $\mathcal{D}^+ = \{(x, y_w)\}$ and $\mathcal{D}^- = \{(x, y_l)\}$. The optimization objective involves learning parameters θ of the policy $\pi_\theta(\mathbf{y}|x)$, which is initialized from a reference policy π_{ref}, to achieve human preference alignment in generated responses.

DPO. (Rafailov et al., 2023) proposed a preference alignment framework that does not require explicit reward modeling. It utilizes the Bradley-Terry (Bradley and Terry, 1952) preference model to derive a closed-form solution for the optimal policy and establishes a mapping relationship from policy parameters to implicit rewards, i.e., $r_\theta(x, y) = \beta \cdot \log\left[\pi_\theta(y \mid x)/\pi_{\text{ref}}(y \mid x)\right]$. Ultimately, it directly optimizes the policy by maximizing the likelihood of preference data.

$$\mathcal{L}_{\text{DPO}}(\theta) = -\mathbb{E}_{\mathcal{D}}\left[\log \sigma\left(r_\theta\left(x, y_w\right) - r_\theta\left(x, y_l\right)\right)\right]. \tag{1}$$

KTO. KTO (Ethayarajh et al., 2024) is a preference alignment method for binary signals. Inspired by prospect theory in economics, KTO decouples the losses of positive and negative preferences to enable training with single-label feedback data. The KTO loss with a reference non-negative constant z_0 is defined as follows.

$$\begin{aligned}\mathcal{L}_{\text{KTO}}(\theta) = &\mathbb{E}_{\mathcal{D}^+}\left[1 - \sigma\left(r_\theta(x, y) - z_0\right)\right] \\ &+ \mathbb{E}_{\mathcal{D}^-}\left[1 - \sigma\left(z_0 - r_\theta(x, y)\right)\right]\end{aligned} \tag{2}$$

KLDO. (Haldar et al., 2025) demonstrated that mainstream alignment algorithms such as DPO and KTO essentially correspond to variational estimates of TV divergence. They constructed the objective function KLDO using the DV representation of KL divergence, thereby achieving a clearer

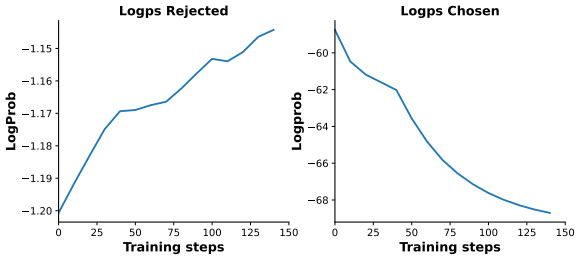


Figure 2: One-sided training in binary preference alignment: monitoring the probability of the opposite side. Training on positives alone raises negative probability, and training on negatives alone raises positive probability.

separation between safe response distributions and harmful response distributions.

$$\mathcal{L}_{\text{KLDO}}(\theta) = -\mathbb{E}_{\mathcal{D}^+}[r_\theta(x,y)] + \ln \mathbb{E}_{\mathcal{D}^-}[e^{r_\theta(x,y)}] \tag{3}$$

Squeezing Effect. (Ren and Sutherland, 2024) conducted a dynamic analysis of each step during the training process and proposed an adverse phenomenon termed the "squeezing effect". They observed that when large negative gradients are applied to already-unlikely outcomes, peak probabilities may be amplified while other probabilities, including those of positive samples, may simultaneously be suppressed.

3.2 Preliminary Experiment

To investigate how training on one side of the preference pair affects the predicted probability of the opposite side in binary preference alignment, we conducted a small experiment with KTO (Ethayarajh et al., 2024).

We first trained the Gemma2-9B-Base (Team et al., 2024) with the UltraChat-200k dataset (Ding et al., 2023) to obtain an SFT model. We then performed binary preference alignment on the UltraFeedback dataset (Cui et al., 2023) using the SFT model as the starting point.

We train using only positive samples while monitoring the probability of negative samples. Intuitively, the probability of negative samples should remain constant or decrease as the probability of positive samples increases. Conversely, when training only with negative samples, the same principle applies. However, as shown in Figure 2, our results indicate that in binary preference alignment training, the probability of the opposite side tends to drift toward an undesired direction.

3.3 Squeezing Effect Extended Claims

To explain the phenomena in Preliminary Experiment 3.2, we draw inspiration from (Ren and Sutherland, 2024) and adopt its proof methodology, extending the three conclusions to the context of binary preference alignment.

For Negative Training:

Claim 1. *The probability of negative sample p_y will decrease, while the probability of peak p_{i^*} will increase.*

Claim 2. *After negative training, p_i with smaller p tends to decrease and vice versa.*

Claim 3. *When the probability of negative sample is extremely small and a peak occurs, except for the peak probability p_{i^*}, which will increase, all other probabilities p_i will decrease.*

For Positive Training:

Extended Claim 1. *The probability of positive sample p_y will increase, while the probability of peak p_{i^*} will decrease.*

Extended Claim 2. *After positive training, p_i with smaller p tends to increase and vice versa.*

Extended Claim 3. *When the probability of positive sample is extremely small and a peak occurs, except for the peak probability p_{i^*}, which will decrease, all other probabilities p_i will increase.*

The proof is in Appendix A.1. It is obvious that both **Claim 3** and **Extended Claim 3** are unacceptable during training. When the **Claim 3** trend occurs, all probabilities are pushed toward the peak, which is detrimental to encouraging positive samples. Similarly, when the **Extended Claim 3** trend occurs, probabilities flowing out from the peak tend to scatter in all directions, which is also detrimental to penalizing negative samples. Therefore, in binary preference alignment training, beyond the direct training effect on the samples themselves, the squeezing effect on the opposite side must also be taken into account. Achieving a proper balance between these two effects is crucial for training efficiency.

3.4 The Learning Objective of PFO

Next, we will introduce in detail the proposed optimization method, which alleviates the squeezing effect while preserving the direct training effect.

To mitigate the squeezing effect and thus more effectively guide model optimization, we propose

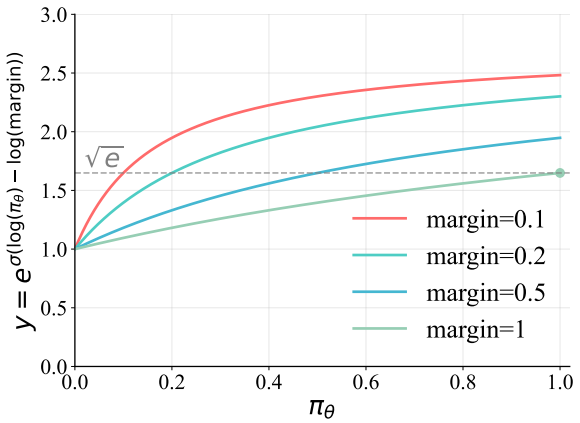


Figure 3: If π_θ is designed directly as r_θ, i.e., $\log(\text{margin}) = 0$, the weight coefficient will have a very small range $[1, \sqrt{e}]$.

to enhance the weight coefficients of samples with larger r_θ. Specifically, this objective is achieved through the following optimization function:

$$\mathcal{L}_{\text{PFO}}(\theta) = -\ln \mathbb{E}_{\mathcal{D}^+}\left[e^{\sigma(r_\theta(x,y))}\right] + \ln \mathbb{E}_{\mathcal{D}^-}\left[e^{\sigma(r_\theta(x,y))}\right] \tag{4}$$

Initially, we considered directly using π_θ as the reward function r_θ to provide an intuitive weight. However, as shown in Figure 3, the weight variation range under this setting was limited, failing to achieve the expected reweighting effect. We need to add some margin to the design of r_θ.

Consistent with (Ren and Sutherland, 2024), we find that in binary preference alignment, the positive squeezing effect is weaker than its negative counterpart. The probability of positive samples also tends to decrease in most cases. Based on the observation, we propose the following inference: For negative samples, if $\pi_\theta - \pi_{\text{ref}}$ is large, it indicates that the current policy is not penalizing enough, and therefore the penalty should be further strengthened. As for positive samples, the large difference indicates that the probability was less affected by the squeezing effect, and according to **Extended Claim 2**, should have a higher likelihood value $\pi_\theta(y|x)$. Therefore, we ultimately chose to use the difference between π_θ and π_{ref} as the reward function r_θ.

3.5 Analysis and Discussion

In this section, we provide an analysis of the PFO optimization objective, including gradient analysis and divergence analysis, to further understand how our PFO balances training effects and squeezing effects.

3.5.1 Gradient Analysis

We compare the gradient analysis of the optimization objectives KTO, KLDO, and PFO. First, we analyze the gradient of the proposed PFO and investigate how it alleviates the squeezing effect:

$$\nabla_\theta \mathcal{L}_{\text{PFO}}(\theta) = -\frac{\mathbb{E}_{\mathcal{D}^+}\left[\nabla_\theta \sigma(r_\theta)\, e^{\sigma(r_\theta)}\right]}{\mathbb{E}_{\mathcal{D}^+}\left[e^{\sigma(r_\theta)}\right]} + \frac{\mathbb{E}_{\mathcal{D}^-}\left[\nabla_\theta \sigma(r_\theta)\, e^{\sigma(r_\theta)}\right]}{\mathbb{E}_{\mathcal{D}^-}\left[e^{\sigma(r_\theta)}\right]} \tag{5}$$

In the proposed PFO method, the optimization weights of positive and negative samples are dynamically adjusted based on their values of $\sigma(r_\theta)$, i.e., $\sigma(\beta \cdot \log[\pi_\theta(y \mid x)/\pi_{\text{ref}}(y \mid x)])$. When the divergence between π_θ and π_{ref} is large, the contribution of the gradient within a batch increases, thereby alleviating the squeezing effect. Notably, we augment r_θ with a sigmoid function $\sigma(\cdot)$. This design primarily prevents samples with exceptionally large r_θ values from dominating the exponential weighting scheme, which would otherwise obscure the contributions of samples with smaller r_θ values and impair their direct training effect. This simple modification effectively balances the constraint on the squeezing effect while preserving the training effect. Furthermore, the sigmoid function not only mitigates the issue of sample suppression but also helps stabilize the gradient dynamics throughout training.

For comparative analysis, we examine the gradient characteristics of KTO and KLDO:

$$\nabla_\theta \mathcal{L}_{\text{KTO}}(\theta) = -\mathbb{E}_{\mathcal{D}^+}\left[\nabla_\theta \sigma(r_\theta - z_0)\right] + \mathbb{E}_{\mathcal{D}^-}\left[\nabla_\theta \sigma(r_\theta - z_0)\right] \tag{6}$$

$$\nabla_\theta \mathcal{L}_{\text{KLDO}}(\theta) = -\mathbb{E}_{\mathcal{D}^+}\left[\nabla_\theta r_\theta\right] + \frac{\mathbb{E}_{\mathcal{D}^-}\left[\nabla_\theta r_\theta\, e^{r_\theta}\right]}{\mathbb{E}_{\mathcal{D}^-}\left[e^{r_\theta}\right]} \tag{7}$$

Gradient analysis shows that KTO does not distinguish between different samples. When the sample probability is low, training may cause the squeezing effect, resulting in insufficient protection of positive sample probabilities and insufficient penalty for negative sample probabilities.

Gradient analysis of KLDO reveals that it also introduces an adaptive weighting mechanism based on r_θ in the negative penalty term. However,

KLDO has two key flaws. First, it lacks a corresponding contribution adjustment mechanism for positive samples, which may lead to insufficient preventive optimization for the trend described in **Extended Claim 3**, which would weaken the penalty for negative samples. Secondly, it implements weight distribution through an exponential function e^{r_θ} of the unbounded variable r_θ. When samples with extremely large r_θ values exist in a batch, the exponential amplification effect can lead to significant weight differences between samples, significantly suppressing or even overwhelming the contributions of some samples. While KLDO may alleviate the squeezing effect, it also significantly reduces the training effect. Even more critically, KLDO's gradient strategy is very aggressive, making training unstable.

3.5.2 Divergence Analysis

Next we will illustrate the relationship between the proposed PFO and the divergence between positive and negative samples from a distribution perspective.

Theorem 3.1. *The maximization objective of PFO constitutes a variational lower bound for the Rényi α-divergence of order $\alpha = 2$ between the positive and negative samples.*

$$\begin{aligned}\mathcal{D}_2(\mathcal{P} \parallel \mathcal{Q}) &\triangleq \log \sum_x \frac{P^2(x)}{Q(x)} \\ &= \sup_T \left\{ \log \mathbb{E}_{v\sim P}\left[e^{T(v)}\right] - \log \mathbb{E}_{v\sim Q}\left[e^{T(v)}\right] \right\}\end{aligned} \tag{8}$$

The proof is provided in Appendix A.2. It is worth noting that when measuring the difference in probability mass distributions of language models, the degree of fit in high-probability regions should be prioritized. Compared to the KL divergence, the Rényi $\mathcal{D}_2$ divergence characterizes the difference in the second-order moments between distributions. The quadratic term $P^2(x)$ in its definition has a significant suppressive effect on the contribution of low-probability events. This property allows the $\mathcal{D}_2$ divergence to more precisely reflect differences in high-probability regions (i.e., the main distribution), while being more robust to small changes in low-probability regions.

3.6 PFO Positive

As discussed above, PFO aims to mitigate the positive and negative squeezing effect in binary preference alignment by applying adaptive weights. In some practical binary preference training scenarios, negative samples do not mean absolute rejection because binary labels lack paired sample references. For example, in a recommendation system, an unclicked option is not necessarily rejected. In such individual feedback settings, positive samples of interest carry substantially greater significance than negative samples of disinterest.

In this context, training should place less emphasis on penalizing negative samples and more on encouraging positive samples while maintaining preference differentiation. Based on the decomposition of the positive and negative squeezing effects, we achieved a balance between the direct effects of training and the squeezing effect within our framework. Specifically, we choose to mitigate the squeezing effect exclusively on negative samples. We designate this PFO variant as PFOP (PFO Positive).

$$\begin{aligned}\mathcal{L}_{\text{PFOP}}(\theta) = &- \mathbb{E}_{\mathcal{D}^+}\left[\sigma(r_\theta(x,y))\right] \\ &+ \ln \mathbb{E}_{\mathcal{D}^-}\left[e^{\sigma(r_\theta(x,y))}\right]\end{aligned} \tag{9}$$

It is worth noting that the optimization objective of PFOP is similar to that of KLDO; however, the key distinction lies in the incorporation of the sigmoid function. This effectively mitigates the issues of sample overwhelming and gradient explosion during training, thereby ensuring training stability and efficiency.

4 Experiments

We designed and conducted two types of experiments to evaluate the effectiveness of the proposed methods. The first set of experiments focuses on the evaluation of multi-turn dialogue capabilities. We use a completely open-source pipeline to train the models and conduct a comprehensive evaluation and comparative analysis on benchmarks such as MT-Bench (Zheng et al., 2023). The second set of experiments focuses on practical binary preference alignment scenarios. Using recommendation systems as a representative example, we evaluate the performance of PFO and its variant PFOP in real-world settings where negative labels are inherently ambiguous due to individual subjective differences in authentic online feedback. The experimental setup for both types of experiments is detailed in Appendix B.1.

Table 1: AlpacaEval 2 (Li et al., 2023) and MT-Bench (Zheng et al., 2023) results under the four settings. LC and WR denote length-controlled and raw win rate, respectively. Our PFO and PFOP can achieve better performance across various settings. PFOP is a variant of PFO that only focuses on increasing the positive probability.

Benchmark	**Method**	**Llama3-8B**		**Gemma2-9B**		**Avg**
		Instruct	**Base**	**Instruct**	**Base**	
MT-Bench	baseline	7.26	5.94	7.76	6.27	6.81
	KTO	7.53	6.57	8.00	**7.19**	7.32
	KLDO	7.28	6.36	7.93	7.11	7.17
	PFOP	**7.72**	6.62	7.89	7.03	7.32
	PFO	7.62	**6.88**	**8.13**	7.00	**7.41**
AlpacaEval 2 (WR/LC)	baseline	32.72/34.11	2.36/3.78	36.02/47.06	4.71/7.14	18.95/23.02
	KTO	36.69/39.73	11.17/15.68	44.92/50.25	19.83/29.00	28.15/33.67
	KLDO	**38.11/41.02**	6.14/9.32	43.84/50.43	8.12/12.70	24.05/28.37
	PFOP	32.72/34.11	12.25/16.49	44.05/**52.41**	18.31/27.38	26.83/32.60
	PFO	36.96/40.41	**14.35/20.56**	**45.06**/50.57	**20.10/31.01**	**29.12/35.64**

Table 2: MovieLens-25M (Harper and Konstan, 2015) and Goodreads (Wan et al., 2019) results.

Metric	**Model**	**Goodreads**	**MovieLens**	**Avg**
HitRate (@5/@10)	KTO	0.020/0.029	0.028/0.033	0.024/0.031
	KLDO	0.022/0.023	0.029/0.036	0.022/0.030
	PFOP	0.026/0.038	**0.033/0.043**	**0.030/0.040**
	PFO	**0.033/0.042**	0.021/0.028	0.027/0.035
DivRatio	KTO	0.1207	0.2223	0.1715
	KLDO	0.1423	0.1567	0.1495
	PFOP	0.2149	0.1947	0.2048
	PFO	**0.2227**	**0.2427**	**0.2327**

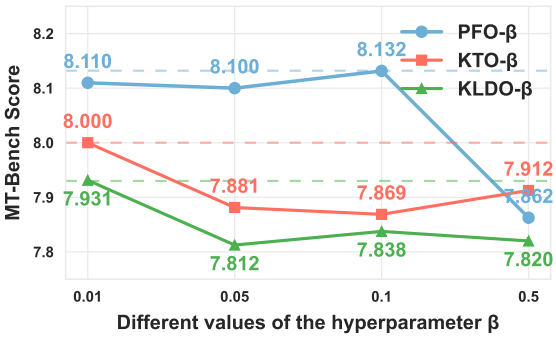


Figure 4: Scores of training models with different hyperparameter β on MT-Bench.

4.1 Dialogue Experiment

Results On Dialogue Benchmarks

As shown in Table 1, we report the performance of the evaluated algorithms on the multi-turn dialogue benchmarks MT-Bench and AlpacaEval 2. We compute the average performance of each algorithm on the four different models across both benchmarks. Our proposed algorithm PFO outperforms KTO and KLDO on both MT-Bench and AlpacaEval 2. Notably, we also evaluate the PFO variant PFOP. As described in Section 3.6, PFOP is less effective than PFO because it only mitigates the squeezing effect of negative training. However, compared to KLDO, which also alleviates the one-sided suppression effect, PFOP better preserves the benefits of direct training and achieves superior results on both MT-Bench and AlpacaEval 2. This demonstrates the effectiveness of introducing the sigmoid function, which mitigates the issue of sample overwhelming during weight assignment and stabilizes the gradients.

To verify the robustness of the algorithm, we report the performance of Gemma2-9B-Instruct under different β on MT-Bench in Figure 4, which demonstrates that PFO optimization retains a relatively robust advantage.

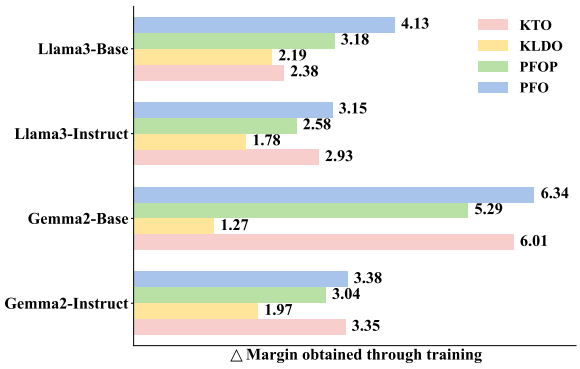


Figure 5: The growth of log probability margin in training samples.

Likelihood Analysis of Training

To verify whether probability flows more efficiently from negative samples to positive samples as hypothesized, we report the Δ-margin (improvement of the difference between the log probabilities of positive and negative samples) for each of the four training methods relative to the original base model. Higher values indicate that the model more effectively rewards positive probabilities and penalizes negative probabilities during training.

Specifically, we randomly sampled 1,000 training samples to test the improvement of the model log-probability margin before and after training. As shown in Figure 5, our proposed algorithm PFO achieves greater improvements in margins across all four models compared to KLDO and KTO.

This indicates that PFO achieves an effective balance between mitigating the squeezing effect and providing protective training benefits. The figure shows that our proposed variant PFOP achieves a larger margin than KLDO, consistent with the observations in Table 1. Furthermore, the log-likelihood margins of both PFOP and KLDO remain lower than those of KTO. This finding further emphasizes that, beyond counteracting the squeezing effect, protecting the training effect is equally essential, underscoring the importance of balancing these two effects in binary preference alignment training.

In Appendix C, we present the one-sided training curves mentioned in the Preliminary Experiment (Section 3.2), shown as Figure 7, together with a more detailed analysis. It can be observed that PFO effectively reduces the squeezing effect compared to KTO. In contrast, KLDO adopts an overly aggressive gradient strategy, which leads to unstable training and poor performance.

4.2 Recommendation Experiment

As shown in Table 2, we report the performance of the evaluated algorithms on the recommendation task on the MovieLens-25M and Goodreads datasets.

From the perspective of HitRate, both of our proposed algorithms PFO and PFOP outperform KTO and KLDO. The further improvement of PFOP over PFO confirms that in scenarios such as recommendation systems, where negative samples are ambiguous, PFOP more effectively maintains positive-sample probabilities while preserving preference discrimination.

In terms of diversity, both of our proposed methods outperform KTO and KLDO, demonstrating that our approach better preserves the diversity of the overall distribution. The advantage of PFOP lies in its ability to mitigate the negative squeezing effect. It counteracts the narrowing trend where high-probability samples increase while low-probability samples decrease, thereby effectively improving diversity. Notably, PFOP selects from the positive samples those that better fit the user's state, which encourages user-specific recommendations. Such an approach may reduce the hit rate but enhances the diversity of global recommendations.

5 Conclusion

This paper examines binary preference alignment optimization from the perspective of probability flow. The inherent non-pairwise nature of binary optimization and the ambiguity of negative samples can induce undesirable probability flow. Through preliminary experiments and theoretical analysis, we identify that the squeezing effect has a non-negligible negative impact on probability flow in binary preference alignment. We propose a framework to mitigate the squeezing effect, in which the balance between the squeezing effect and the direct training effect can be adjusted according to the degree of negative-sample ambiguity. Extensive experiments on benchmarks and datasets demonstrate that our proposed methods PFO and PFOP significantly outperform existing state-of-the-art approaches. We hope these findings offer new insights into the squeezing effect and guide future improvements in binary preference alignment methods.

Limitations

Positive-Only Optimization for the Squeezing Effect. We have developed a binary optimization method (PFO) to address the squeezing effect, along with a negative-focused variant (PFOP) designed for the common negative-label ambiguity scenario in binary preference alignment. However, we have not separately evaluated the effect of positive-only optimization, as we have not yet identified a concrete scenario in which the primary emphasis is placed on penalizing negative samples. Nonetheless, such scenarios may exist in practice. We therefore leave the investigation of positive-only optimization for future work.

More Optimized Weight Representation. We propose a method to mitigate the squeezing effect by adjusting sample weights, where the weight modulation term is $\pi_\theta - \pi_{\text{ref}}$. As mentioned in Section 3.4, this design aims to provide a larger dynamic range for the weights. However, we have not yet explored this direction extensively. Other formulations may exist that better balance the trade-off between mitigating the squeezing effect and preserving the direct training effect. We leave the investigation of more optimized schemes for future work, and hope that the dynamic impact of one-sided sample training on binary preference alignment can attract more attention.

Ethics Statement

We adhered to the open-source model usage agreement, using the language model solely for scientific research purposes. The datasets we used were all open-source datasets and were commonly used in academia. These datasets do not contain any information that could identify personal names or contain offensive content.

During the writing process, we used the translation tool DeepL to assist with the translation. This tool may offer some LLM-based AI translation. Nevertheless, we carefully reviewed the translated content to ensure that the translation did not alter our intended meaning.

References

Kingma DP Ba J Adam and 1 others. 2014. A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 1412(6).

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, and 1 others. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

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

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. *Preprint*, arXiv:2310.01377.

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. *Preprint*, arXiv:2305.14233.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

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

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

Chongming Gao, Ruijun Chen, Shuai Yuan, Kexin Huang, Yuanqing Yu, and Xiangnan He. 2025. Sprec: Self-play to debias llm-based recommendation. In *Proceedings of the ACM on Web Conference 2025*, pages 5075–5084.

Rajdeep Haldar, Ziyi Wang, Qifan Song, Guang Lin, and Yue Xing. 2025. Llm safety alignment is divergence estimation in disguise. *arXiv preprint arXiv:2502.00657*.

F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19.

Mojan Javaheripi, Sébastien Bubeck, Marah Abdin, Jyoti Aneja, Sebastien Bubeck, Caio César Teodoro Mendes, Weizhu Chen, Allie Del Giorno, Ronen Eldan, Sivakanth Gopi, and 1 others. 2023. Phi-2:

The surprising power of small language models. *Microsoft Research Blog*, 1(3):3.

Haozhe Ji, Cheng Lu, Yilin Niu, Pei Ke, Hongning Wang, Jun Zhu, Jie Tang, and Minlie Huang. 2024a. Towards efficient exact optimization of language model alignment. *arXiv preprint arXiv:2402.00856*.

Jianchao Ji, Zelong Li, Shuyuan Xu, Wenyue Hua, Yingqiang Ge, Juntao Tan, and Yongfeng Zhang. 2024b. Genrec: Large language model for generative recommendation. In *European Conference on Information Retrieval*, pages 494–502. Springer.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. *arXiv preprint arXiv:2306.02561*.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. 2024. Vineppo: Unlocking rl potential for llm reasoning through refined credit assignment. *arXiv preprint arXiv:2410.01679*.

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

Yaxuan Li and 1 others. 2026. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*.

Chris Yuhao Liu, Liang Zeng, Jiacai Liu, Rui Yan, Jujie He, Chaojie Wang, Shuicheng Yan, Yang Liu, and Yahui Zhou. 2024. Skywork-reward: Bag of tricks for reward modeling in llms. *arXiv preprint arXiv:2410.18451*.

Xin Mao, Feng-Lin Li, Huimin Xu, Wei Zhang, Wang Chen, and Anh Tuan Luu. 2024. As simple as fine-tuning: Llm alignment via bidirectional negative feedback loss. *arXiv preprint arXiv:2410.04834*.

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.

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*.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, and 1 others. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, and 1 others. 2025. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*.

Qwen Team. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

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*.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. 2020. TRL: Transformers Reinforcement Learning.

Mengting Wan, Rishabh Misra, Ndapa Nakashole, and Julian McAuley. 2019. Fine-grained spoiler detection from large-scale review corpora. *arXiv preprint arXiv:1905.13416*.

Binghai Wang, Rui Zheng, Lu Chen, Yan Liu, Shihan Dou, Caishuang Huang, Wei Shen, Senjie Jin, Enyu Zhou, Chenyu Shi, and 1 others. 2024. Secrets of rlhf in large language models part ii: Reward modeling. *arXiv preprint arXiv:2401.06080*.

Teng Xiao, Yige Yuan, Zhengyu Chen, Mingxiao Li, Shangsong Liang, Zhaochun Ren, and Vasant G Honavar. 2025. Simper: A minimalist approach to preference alignment without hyperparameters. *arXiv preprint arXiv:2502.00883*.

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. 2025. Learning to reason under off-policy guidance. *arXiv preprint arXiv:2504.14945*.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2025a. Dapo:

An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

Zhuohao Yu, Jiali Zeng, Weizheng Gu, Yidong Wang, Jindong Wang, Fandong Meng, Jie Zhou, Yue Zhang, Shikun Zhang, and Wei Ye. 2025b. Rewardanything: Generalizable principle-following reward models. *arXiv preprint arXiv:2506.03637*.

Yu Yue, Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, TianTian Fan, Zhengyin Du, and 1 others. 2025. Vapo: Efficient and reliable reinforcement learning for advanced reasoning tasks. *arXiv preprint arXiv:2504.05118*.

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.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, and 1 others. 2023. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36:55006–55021.

A Proof

A.1 Proof of Claims

Proof. For ease of understanding, we follow the notation of Ren and Sutherland (2024). The ratio of change in the confidence of each token i is given by

$$\alpha_i \triangleq \frac{p_i^{t+1}}{p_i^t} = \frac{\sum_{j=1}^{V} e^{z_j^t}}{\sum_{j=1}^{V} \beta_j e^{z_j^t}}. \tag{10}$$

Here p_i^t denotes the probability of a particular token i, while p_i^{t+1} represents the probability after one training step; z denotes the logits of the policy and V the size of the vocabulary. Note that the value of β_j depends on whether token i is the trained sample y, with Case 1 ($i = y$) and Case 2 ($i \neq y$):

$$\begin{aligned} \text{Case 1: } \beta_j &= \begin{cases} e^{-\eta'(1+p_j^t-p_i^t)} & \text{if } j \neq y \\ 1 & \text{if } j = y \end{cases} \\ \text{Case 2: } \beta_j &= \begin{cases} e^{-\eta'(p_j^t-p_i^t)} & \text{if } j \neq y \\ e^{-\eta'(p_j^t-p_i^t-1)} & \text{if } j = y \end{cases} \end{aligned} \tag{11}$$

Where η' is the learning rate, which can be negative if negative gradients are considered during training, and vice versa.

Extended Claim 1. *The probability of positive sample p_y will increase, while the probability of peak p_{i^*} will decrease.*

Proof. For positive sample optimization, denote $p_{i^*} = \arg\max_{i \in [V] \setminus \{y\}} p_i$ as the peak probability other than target sample y. In Case 1 ($\beta \leq 1$), the scaling factor satisfies $\alpha > 1$, inducing an increase in p_i. In Case 2, for the non-target class ($j \neq y$) the condition $p_j \leq p_{i^*}$ implies $\beta \geq 1$, while for the target class ($j = y$) $\beta > 1$ holds strictly, collectively yielding $\alpha < 1$ and consequently a decrease in p_i.

Extended Claim 2. *After positive training, p_i with smaller p tends to increase and vice versa.*

Proof. Now we observe p_i' and perform a qualitative analysis. Consider the partition of classes into two subsets: $\mathcal{B} = \{j : p_j > p_i'\}$ and $\mathcal{S} = \{j : p_j \leq p_i'\}$. In Case 2, as p_i increases, the contribution of terms with $\beta_j > 1$ in $\mathcal{S}$ dominates, increasing the probability that $\alpha < 1$ and thus inducing a decrease in p_i. Conversely, at lower p_i values, terms with $\beta_j < 1$ in $\mathcal{B}$ become dominant, driving an increase in p_i.

Extended Claim 3. *When the probability of positive sample is extremely small and a peak occurs, except the peak probability p_{i^*} decreasing, all other probabilities p_i will increase.*

Proof. When a peaky target probability p_{i^*} emerges, the corresponding logit z_{i^*} is also very large. Under this condition, the value of α is dominated by β_{i^*}, i.e., for the observed $\alpha_{i'} \approx \frac{e^{z_{i^*}}}{\beta_{i^*} e^{z_{i^*}}} = \frac{1}{\beta_{i^*}} > 1$. Consequently, the observed probability $p_{i'}$ is dominated by β_{i^*} so that $\alpha_{i'} > 1$. By applying **Extended Claim 1**, when the peak probability does not correspond to the positive-sample class, this induces a decrease in the peak probability while simultaneously increasing all other class probabilities.

A.2 Proof of Theorem 3.1

Proof. To prove Theorem 3.1, we need to prove $\frac{E_P[e^T]}{E_Q[e^T]} \leq E_Q\left[\left(\frac{dP}{dQ}\right)^2\right]$.

We first denote $f = \frac{e^T}{E_Q[e^T]}$. Then we obtain

$$\begin{cases} E_Q[f] = 1, \\ \dfrac{E_P[e^T]}{E_Q[e^T]} = E_P[f]. \end{cases} \tag{12}$$

Our objective is to show that the maximum value of $\frac{E_P[e^T]}{E_Q[e^T]} = E_P[f]$ is $E_Q\left[\left(\frac{dP}{dQ}\right)^2\right]$, so we only need to find the maximum value of $E_P[f]$ under the constraint $E_Q[f] = 1$. Using the Lagrange multiplier method, let

$$\mathcal{L}_f(v) = E_P[f] - \lambda(E_Q[f] - 1) \tag{13}$$

$$= \int p(v)f(v)dv - \lambda\left(\int q(v)f(v)dv - 1\right) \tag{14}$$

Setting the derivative of $\mathcal{L}$ to zero, we can get

$$\frac{\delta \mathcal{L}}{\delta f(v)} = p(v) - \lambda q(v) = 0 \implies f(v) = \frac{p(v)}{\lambda q(v)} \tag{15}$$

Substituting into the constraint $\int q(v)f(v)dv = 1$, we can get

$$\int q(v) \cdot \frac{p(v)}{\lambda q(v)}\, dv = 1 \tag{16}$$

$$\implies \frac{1}{\lambda}\int p(v)\, dv = 1 \tag{17}$$

$$\implies \frac{1}{\lambda} = 1 \tag{18}$$

$$\implies \lambda = 1 \tag{19}$$

Substituting back into $f(v) = \frac{p(v)}{\lambda q(v)}$, the maximum of $E_P[f]$ is

$$\frac{E_P[e^T]}{E_Q[e^T]} = E_P[f] \leq \int p(v) \frac{p(v)}{q(v)} dv \tag{20}$$

$$= \int p(v) \frac{p(v)}{q(v)} \cdot \frac{q(v)}{q(v)} dv \tag{21}$$

$$= E_Q\left[\left(\frac{dP}{dQ}\right)^2\right]. \tag{22}$$

Taking the logarithm of both sides concludes the proof of Theorem 3.1.

A.3 Discussion on the Variational Gap

To alleviate the sample drowning problem, we introduce a sigmoid function as a nonlinear mapping in the parameterization of $T(v)$. The constraints imposed by this function on the output values may make the theoretical upper bound of PFO unattainable in practice, which could in turn affect the tightness of the variational bound. To examine the practical impact of this issue, we conducted multiple hyperparameter experiments (see Section 4 for details) and found that performance remained stable across multiple trials. A more rigorous theoretical characterization of this variational gap is left for future work.

B Experiment Details

B.1 Dialogue Experiment

Models and Datasets. We performed alignment with two families of open-source models, Llama3-8B-Base/Instruct (Dubey et al., 2024) and Gemma2-9B-Base/Instruct (Team et al., 2024).

For the **Base** setup, we followed the training pipeline of SIMPO (Meng et al., 2024) and Zephyr (Tunstall et al., 2023). We first trained the base model on the UltraChat-200k dataset (Ding et al., 2023) to obtain an SFT model. We then performed binary preference alignment on the UltraFeedback dataset (Cui et al., 2023), using the SFT model as the starting point.

For the **Instruct** setup, we use off-the-shelf instruction-tuned models, i.e., meta-llama/Meta-Llama-3-8B-Instruct (Dubey et al., 2024) and google/gemma-2-9b-it (Team et al., 2024). In order to obtain better training data, we reused the training datasets princeton-nlp/llama3-ultrafeedback-armorm and princeton-nlp/gemma2-ultrafeedback-armorm processed by (Meng et al., 2024). For each prompt in UltraFeedback dataset (Cui et al., 2023), they generated 5 responses using the SFT model. These 5 responses are then scored using llm-blender/PairRM (Jiang et al., 2023), and the response with the highest score is selected as chosen and the response with the lowest score is selected as rejected.

Evaluation Benchmarks. We primarily evaluate our models using two of the most popular open-source instruction-following benchmarks: MT-Bench (Zheng et al., 2023) and AlpacaEval 2 (Li et al., 2023). These benchmarks assess models' multi-dimensional conversational capabilities across multiple topics and have been widely adopted by the community. MT-Bench covers 80 questions across 8 categories. AlpacaEval 2 contains 805 questions from 5 datasets. For MT-Bench, we report the average MT-Bench score of models evaluated by GPT-4o. For AlpacaEval 2, we report the raw win rate (WR) and length-controlled win rate (LC) (Dubois et al., 2024) of models evaluated against GPT-4-Preview-1106 using GPT-4o as the judge. The LC metric is specifically designed to be robust to the effects of model length.

Baselines. We compare PFO and PFOP with the binary preference optimization methods KTO (Ethayarajh et al., 2024) and KLDO (Haldar et al., 2025). Considering that hyperparameter tuning is crucial to achieve the best performance of preference optimization methods, we thoroughly tuned the hyperparameters for each baseline and reported the best performance.

B.2 Recommendation Experiment

Model and Datasets. We evaluate a binary preference recommendation system for next-query prediction using each user's historical queries, built on top of the open-source Qwen2.5-Instruct-7B model (Team, 2024). The experiments use the MovieLens-25M (Harper and Konstan, 2015) and Goodreads (Wan et al., 2019) datasets. The datasets employ real-world user logs, where training and test partitions are temporally segmented. Query examples are shown in Table 4. Following the data processing approach in (Ji et al., 2024b) and (Gao et al., 2025), for each user interaction sequence, the terminal element constitutes test data while preceding elements form train data. Rejected samples are queries from other users and different from the chosen sample.

Metrics. We evaluate model performance using HitRate and DivRatio. The HitRate metric is

commonly used in recommendation systems and indicates the percentage of items recommended by the model that match those in the ground-truth data. The DivRatio metric is defined as the proportion of unique recommendations, reflecting the diversity of the recommendations. Note that during evaluation we do not filter the model outputs for validity but simply ignore the content in brackets, since we contend that the ability to precisely generate correct item names is a key indicator of a model's proficiency.

B.3 Training Hyperparameters

For general hyperparameters, we applied the following setup: We used a batch size of 128, a maximum sequence length of 8192, and a cosine learning rate schedule with a 10% warmup step per epoch, with all implemented using the Adam optimizer (Adam et al., 2014). The meaning of the hyperparameter β varied across the different baseline methods, and the corresponding search strategies are shown in Table 3. Each method independently searched for a learning rate in the range [5e-7, 8e-7, 1e-6]. For the binary preference learning algorithm, the positive and negative weights can be separated and adjusted separately. In this paper, the default value $\lambda_l = \lambda_w = 1.0$.

Our experiments are based on the standard implementation provided by trl (von Werra et al., 2020). The dialogue experiment trained for approximately 8 hours on 8 A800 GPUs, and the recommendation experiment training time is approximately 3 hours on 8 A800 GPUs.

Table 3: Hyperparameter Search Range for Different Methods

Method	Hyperparameter Search Range
KTO	$\begin{cases} \lambda_l = \lambda_w = 1.0 \\ \beta \in [0.01, 0.05, 0.1] \end{cases}$
KLDO	$\begin{cases} \lambda_l = \lambda_w = 1.0 \\ \beta \in [0.01, 0.05, 0.1] \end{cases}$
PFOP	$\begin{cases} \lambda_l = \lambda_w = 1.0 \\ \beta \in [0.01, 0.05, 0.1] \end{cases}$
PFO	$\begin{cases} \lambda_l = \lambda_w = 1.0 \\ \beta \in [0.01, 0.05, 0.1] \end{cases}$

C More Experiment Analysis

Here we present further analysis of the experimental process, including the training curves, the one-sided experiments in the Preliminary Experiment, and the monitoring of grad norm.

Training Curves. Figure 6 shows the training curves for Gemma2-9B-Instruct. PFO achieves the largest log probability margin, indicating that it transfers probability mass from negative samples to positive samples to the greatest extent. This demonstrates the effectiveness of mitigating the squeezing effect. Similarly, KLDO also incorporates optimization for the squeezing effect in negative training, yet its probability margin is the lowest. This is due to its aggressive gradient strategy, which leads to unstable training. Moreover, the issue of sample overwhelming further degrades the direct training effect. In contrast, although PFOP only performs one-sided squeezing optimization, its probability margin is comparable to those of KTO and PFO, and it achieves a higher probability for positive samples. This benefit comes from the protection provided by its sigmoid function, a simple design that significantly enhances the direct training effect and training stability.

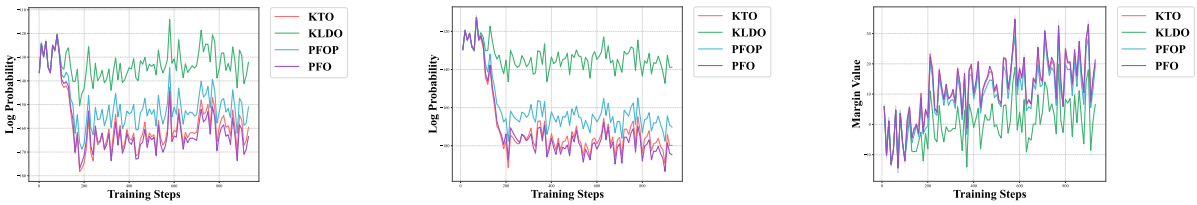
One-sided Experiment. Moreover, we repeated the preliminary experiment by training only one side of the samples in binary preference alignment while monitoring the probability of the opposite side. Specifically, we conducted experiments on both Gemma2-9B-Base and Gemma2-9B-Instruct, and the results are shown in the Figure 7.

In the experiments with the Instruct model, the probability flow during one-sided training was more favorable. When training on positive samples, the probabilities of negative samples decreased (except for KLDO), while the probabilities of positive samples increased. We attribute this difference from the Base model experiments to the alignment between the dataset and the initial model state. In the Base model experiments, the dataset was not directly generated by the model itself, and the sample probabilities were relatively low, which led to a more severe squeezing effect. In contrast, for the Instruct model, the dataset was self-generated by the model, so the probabilities of negative samples were not too low, resulting in a weaker squeezing effect.

Comparing the results on two models, we observe that PFO is less affected by the squeezing effect, indicating that our proposed method can

Interaction history: Independence Day (a.k.a. ID4) (1996),Star Wars: Episode VI - Return of the Jedi (1983),True Lies (1994),Aladdin (1992),Fargo (1996)
Recommendation Prompt Example: **Instruction:** Using the past movie choices, can you predict the next they will likely watch? Independence Day (a.k.a. ID4) (1996),Star Wars: Episode VI - Return of the Jedi (1983),True Lies (1994),Aladdin (1992) **output:** Fargo (1996)

Table 4: Example of Interaction History and Recommendation Prompt



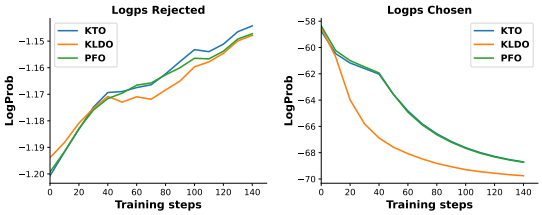
(a) The log probability of chosen during training.

(b) The log probability of rejected during training.

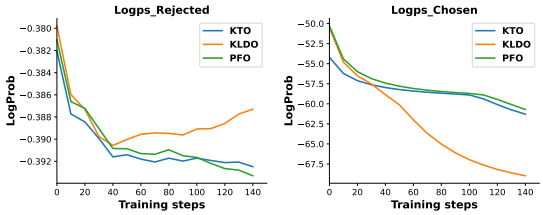
(c) The log probability margin during training.

Figure 6: Training dynamics of KTO, KLDO, PFO and PFOP on Gemma2-9B-Instruct (Team et al., 2024). In binary preference alignment training, the log probability of positive samples also exhibits a decreasing trend, and the proposed PFO algorithm achieves the most significant improvement in the margin between rejected and chosen.

effectively mitigate the squeezing effect. Unexpectedly, KLDO produced disappointing results in most experiments. We attribute this to its overly aggressive gradient strategy, which leads to unstable training. As shown in Figure 8, the gradient norm curve of KLDO during training is highly unstable, exhibiting sharp spikes that persist even in the middle stage of training. This indicates that the unbounded nature of KLDO indeed reduces convergence efficiency and leads to training instability. This phenomenon also confirms the necessity of adding a sigmoid function.



(a) Train only one side of Gemma2-base and monitor the probability changes on the other side.



(b) Train only one side of Gemma2-instruct and monitor the probability changes on the other side.

Figure 7: One-sided training experiment on Gemma2-9B-Base and Gemma2-9B-Instruct, training only one side of the samples and monitoring the probability on the untrained side.

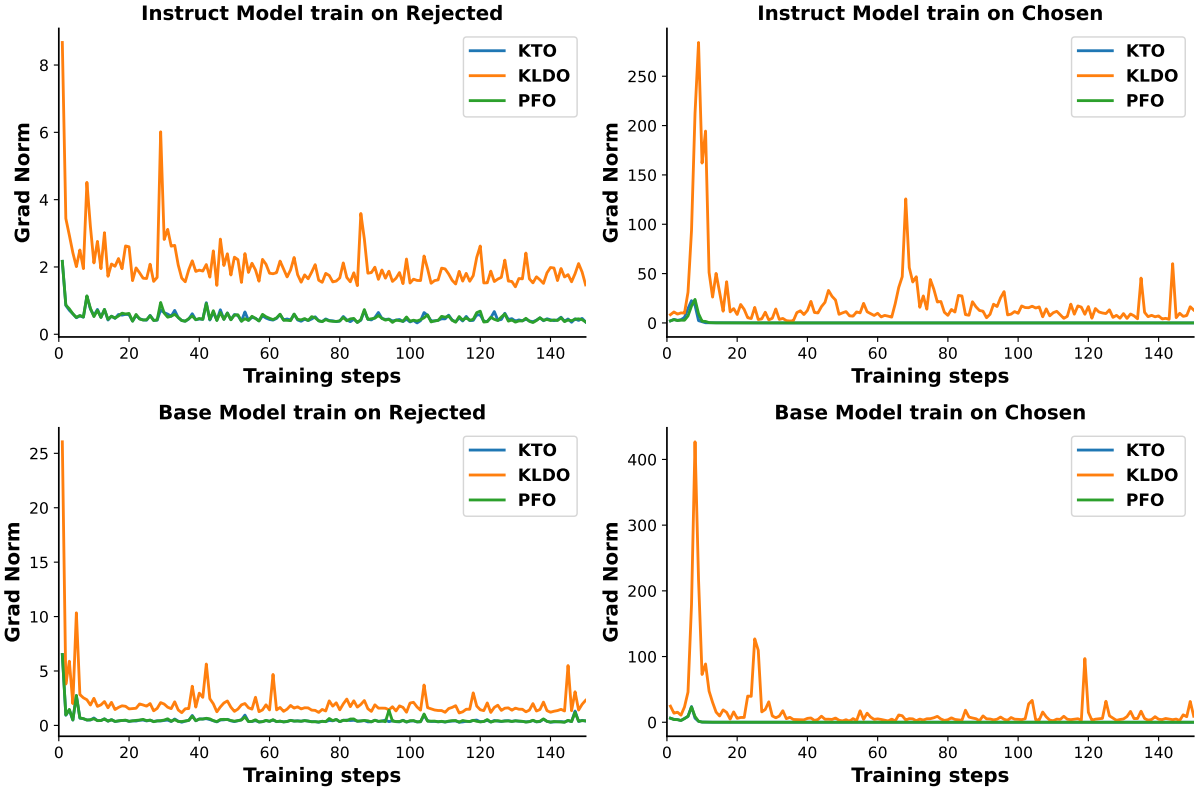


Figure 8: Gradient norm dynamics during one-sided training.