Optimal Transport-Based Token Weighting scheme for Enhanced Preference Optimization

Meng Li[1*], **Guangda Huzhang**[2], **Haibo Zhang**[2], **Xiting Wang**[1†‡], **Anxiang Zeng**[2†]
[1]Gaoling School of Artificial Intelligence, Renmin University of China,
[2]LLM Team, Shopee Pte. Ltd.
mengli.24@ruc.edu.cn, guangda.huzhang@shopee.com, peter.wu@shopee.com
xitingwang@ruc.edu.cn, zeng0118@e.ntu.edu.sg

Abstract

Direct Preference Optimization (DPO) has emerged as a promising framework for aligning Large Language Models (LLMs) with human preferences by directly optimizing the log-likelihood difference between chosen and rejected responses. However, existing methods assign equal importance to all tokens in the response, while humans focus on more meaningful parts. This leads to suboptimal preference optimization, as irrelevant or noisy tokens disproportionately influence DPO loss. To address this limitation, we propose **O**ptimal **T**ransport-based token weighting scheme for enhancing direct **P**reference **O**ptimization (OTPO). By emphasizing semantically meaningful token pairs and de-emphasizing less relevant ones, our method introduces a context-aware token weighting scheme that yields a more contrastive reward difference estimate. This adaptive weighting enhances reward stability, improves interpretability, and ensures that preference optimization focuses on meaningful differences between responses. Extensive experiments have validated OTPO's effectiveness in improving instruction-following ability across various settings.[1]

1 Introduction

Aligning large language models with human preferences (Ouyang et al., 2022) and values (Yao et al., 2023; Yi et al., 2023) guides LLMs to be helpful, honest, and harmless, preventing misuse of their powerful abilities (Bai et al., 2022). Reinforcement Learning from Human Feedback (RLHF) achieves this objective via fine-tuning the LLM to optimize the learned reward model (Ouyang et al., 2022).

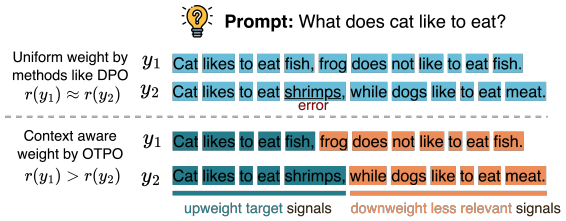


Figure 1: The uniform weighting in DPO leads to suboptimal alignment results, allowing less relevant signals to dominate. OTPO identifies the contextually similar parts in pairwise responses as targets and upweights target signals. $r(y_*)$ denotes the estimated reward under each method.

Offline direct preference optimization algorithms, e.g., DPO (Rafailov et al., 2024b), simplify this process by applying reparameterization to implicitly model the reward as the log ratio likelihood of the response, which is equivalent to the sum of log ratio likelihoods of all tokens. This transformation results in a simple binary cross-entropy objective of reward difference, and has been widely adopted due to its training stability and efficiency (Xiao et al., 2024).

The DPO loss treats each token equally, which can bias the model to overlook less important factors and learn by shortcuts, leading to suboptimal results (Park et al., 2024). In Fig. 1, tokens less relevant to the question dominate the reward, while important parts like "*Cat likes to eat fish*" should be paid more attention. Current methods, including SimPO (Meng et al., 2024), SamPO (Lu et al., 2024), and LDDPO (Liu et al., 2024b), primarily focus on the length bias caused by imbalanced total token weight between a chosen response and a rejected response. They apply a heuristic weighting scheme to reduce the difference in total token weight. Moreover, they can not distinguish the important tokens relevant to instruction-following due to a lack of supervision signal. Recent work like APO (Dao, 2024) has attempted to address this issue by rewriting the irrelevant parts to maintain

[*]Work done during internship at Shopee LLM Team.
[†]Corresponding Authors.
[‡]Work partly done at Beijing Key Laboratory of Research on Large Models and Intelligent Governance and Engineering Research Center of Next-Generation Intelligent Search and Recommendation, MOE.
[1]Code is available at https://github.com/Mimasss2/OTPO.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 21311–21334
July 27 - August 1, 2025

minimum difference with the other response, yet it relies heavily on the external reviser.

In this paper, we propose an **O**ptimal **T**ransport-based weighting scheme for direct **P**reference **O**ptimization (OTPO), a novel unsupervised framework for calculating token weights in direct preference optimization. Our key innovation lies in emphasizing tokens where the responses agree (similar tokens) as indicators of higher quality or shared information, and de-emphasizing tokens where they disagree. We observe that these shared parts of responses are more likely to be relevant to the question, as there are multiple ways to represent the same answer. Specifically, we utilize an unbalanced optimal transport approach to dynamically assign a fixed total weight budget to token-level weights based on the similarity between tokens in chosen and rejected responses, allocating higher weights to more semantically relevant tokens. This allows for estimating the minimum effort required to transform one response to the other. Compared to previous methods, our method improves reward stability, enhances the transparency of preference optimization, and ensures the optimization focuses on more meaningful differences between responses.

To sum up, our contributions are threefold:

- We identify the issue of treating tokens equally in DPO and propose a general weighting scheme that incorporates previous methods.
- We design an optimal transport-based token weighting scheme to identify important tokens without extra supervision signal.
- Extensive experiments have validated OTPO's effectiveness across various settings, achieving up to 10.9% length-controlled win rate increase on AlpacaEval2 compared to DPO.

2 Methods

We first provide a simple background of DPO (Sec. 2.1). To refine the understanding of the reward difference term Δ_r in DPO, we decompose it from a more granular perspective and propose a general weighting scheme that incorporates previous methods (Sec. 2.2). By examining interactions at the level of chosen-rejected token pairs, we identify opportunities for improvement in how reward differences are computed and propose our OTWPO algorithm for more nuanced adjustments (Sec. 2.3). Fig. 2 shows the overall framework.

2.1 Background: Direct Preference Optimization

DPO eliminates the need for explicitly learning a reward model and reparameterizes the reward model as:

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

Here, π_θ is the model to be optimized, π_{ref} is a reference model, $\pi_*(y|x)$ denotes the probability of a response y given input x under policy π_*, and $Z(x)$ is an unknown partition function. Incorporating the above reward model into the Bradley-Terry model, the final DPO loss function is:

$$\mathcal{L}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_c,y_r)\sim D}[\log \sigma(\beta \Delta_r)] \quad (2)$$

$$\Delta_r = \log \frac{\pi_\theta(y_c|x)}{\pi_{\text{ref}}(y_c|x)} - \log \frac{\pi_\theta(y_r|x)}{\pi_{\text{ref}}(y_r|x)} \quad (3)$$

where (x, y_c, y_r) is a preference pair consisting of a prompt x, a chosen response y_c, and a rejected response y_r from the preference dataset D. And σ denotes the sigmoid function.

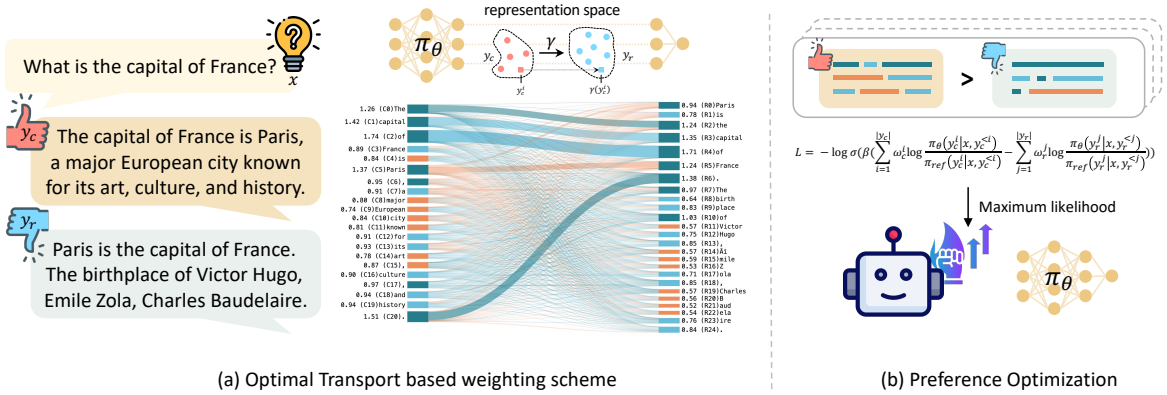
Formally, given a response y of length $|y|$, its probability under the policy is factorized as the multiplicative product of each token's probability $\pi_\theta(y|x) = \prod_{i=1}^{|y|} \pi_\theta(y^i|x, y^{<i})$. The reparameterized reward difference term Δ_r in DPO treats the entire response as a single action, in contrast to classical RLHF methods that model each token as an action and optimize token-level value functions with sparse rewards at the terminal state (Rafailov et al., 2024a). This can mislead optimization by causing the policy to focus on less important tokens and learn through shortcuts, potentially undermining the intended reward signal.

2.2 Decomposing DPO Loss

We first break down the reward difference Δ_r in DPO loss at the token level to explicitly reveal how each token contributes to the optimization process. Operating at the token level, we have (see proofs in Appx. K.1):

$$\Delta_r = \sum_{i=1}^{|y_c|} q_c^i - \sum_{j=1}^{|y_r|} q_r^j, \quad \text{where} \quad q_*^i = \log \frac{\pi_\theta(y_*^i|x, y_*^{<i})}{\pi_{\text{ref}}(y_*^i|x, y_*^{<i})} \quad (4)$$

Here, q_*^i denotes the log-likelihood ratio of the i-th token in y_* between the model and the reference



(a) Optimal Transport based weighting scheme

(b) Preference Optimization

Figure 2: Overall framework. (a) We compute the token-level weighting scheme using optimal transport. Each response's distribution is made up of its tokens, represented as vectors in the LLM's representation space. The optimized transport plan is visualized using a Sankey diagram. (b) We decompose the DPO loss at the token level and apply the weighting scheme obtained in (a).

Method	Weighting Scheme
DPO	$\omega_i = 1,\ \forall i \in [1, \lvert y \rvert]$
SimPO	$\omega_i = \frac{1}{\lvert y \rvert},\ \forall i \in [1, \lvert y \rvert]$
SamPO	$\omega_i = 1,\ \forall i \in S\ ,\quad \omega_i = 0,\ \forall i \notin S$ where $S \sim \text{Uniform}(m, [1, \lvert y \rvert])$
LDDPO	$\omega_i = 1,\ \forall i \in [1, m]$ $\omega_i = \alpha,\ \forall i \in [m+1, \lvert y \rvert]$
OTPO	$\omega_c^i = \sum_{j=1}^{\lvert y_r \rvert} \Gamma_{i,j}\ ,\quad \omega_r^i = \sum_{i=1}^{\lvert y_c \rvert} \Gamma_{i,j}$

Table 1: Weighting schemes for different methods. Here, $|y|$ is the current response length, $|y_c|$ and $|y_r|$ denote the lengths of the chosen and rejected responses, respectively. $m = \min(|y_c|, |y_r|)$, and $\alpha \in [0, 1]$ is a hyperparameter in LDDPO. Γ is the optimized transport plan in Eq. 9.

distribution. We can see that each token contributes equally by its log-likelihood ratio. We further incorporate a token-level weighting scheme for Δ_r and express it as:

$$\Delta_r = \sum_{i=1}^{|y_c|} \omega_c^i q_c^i - \sum_{j=1}^{|y_r|} \omega_r^j q_r^j \tag{5}$$

where ω_*^i represents the weight assigned to the i-th token in response y_*. This decomposition expresses Δ_r in terms of differences in weighted token log-probability ratio difference. DPO can be viewed as a special case of Eq. 5, assigning a uniform weight of 1 for all tokens.

This token weighting scheme incorporates previous methods for mitigating length bias, as shown in Tab. 1. Fig. 3 provides a more intuitive illustration. Length bias refers to the phenomenon where the model learns to only improve length instead of quality to increase reward difference compared to the base model. DPO causes length bias as the difference in the two responses' total token weight $\delta = \sum_{i=1}^{|y_c|} \omega_c^i - \sum_{j=1}^{|y_r|} \omega_r^j = |y_c| - |y_r|$ being positive most of the time. Previous methods essentially apply a weighting scheme to reduce the total token weight difference δ. SimPO down-weight all tokens' weight to $1/|y|$, ensuring each response's total weight sums to 1. SamPO employs a more subtle downsampling on the longer response to only consider the same amount of random tokens as the other response. LDDPO only down-weights the over-lengthy parts to reduce δ. These methods apply some heuristic kind of weighting scheme to mitigate length bias caused by total token weight bias. Yet a more principled weighting scheme delving into each token's importance is needed to solve the problem fundamentally.

2.3 Optimal Transport based Weighting Scheme

While prior methods primarily adjust total token weights, our approach takes a deeper look into the geometric structure of token pair relationships. We further break down the reward difference term as the weighted token log-likelihood difference across all chosen-rejected token pairs:

$$\Delta_r = \sum_i \sum_j \Gamma_{i,j} (q_c^i - q_r^j) \tag{6}$$

Here, $\Gamma_{i,j}$ represents the weight assigned to each token pair $\{y_c^i, y_r^j\}$. Then, the previous token level weights ω_c^i, ω_r^j corresponds to:

$$\omega_c^i = \sum_j \Gamma_{i,j}, \quad \omega_r^j = \sum_i \Gamma_{i,j}. \tag{7}$$

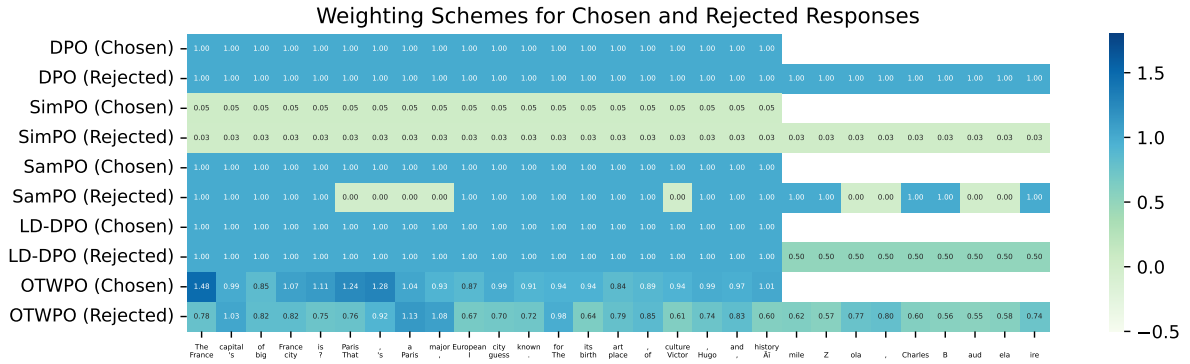


Figure 3: Weights assigned to the responses given different methods. Here, given the prompt *"What is the capital of Paris?"*, the chosen response is *"The capital of France is Paris, a major European city known for its art, culture, and history."*, and the rejected response is *"France's big city? That's Paris, I guess. The birthplace of Victor Hugo, Émile Zola, Charles Baudelaire."*

Building on the token-pair level transformation of the DPO loss, we now focus on exploiting this finer granular Δ_r to differentiate token pairs based on their semantic relevance. We aim to emphasize semantically meaningful token pairs, with similar meanings and structural roles, while de-emphasizing less relevant ones. This coincides with the principle of majority voting, where the most frequently occurring element is chosen as the final answer (Wang et al., 2023). In our context, the parts of the responses most relevant to the question—such as the direct answer "The capital of France is xxx"—are more likely to appear consistently across both the chosen and rejected responses. This is particularly evident in the on-policy setting, where all responses are generated by the same policy. These shared parts, representing the "majority", carry critical information and thus are prioritized.

A key challenge arises from the unequal total weight sums of chosen and rejected responses in naive DPO loss, while the above formulation inherently requires the same total weight sum for each response pair, as:

$$\sum_i \omega_c^i = \sum_j \omega_r^j = \sum_i \sum_j \Gamma_{i,j} \tag{8}$$

The optimal Γ is computed by solving the optimization problem later defined in Equation 9. We ideally want a weighting scheme that accounts for their structural differences while still ensuring fairness in total weight. A natural way to achieve this is through optimal transport, which provides a principled method for aligning distributions while minimizing their discrepancy. However, standard optimal transport assumes equal total mass on both sides, making it incompatible with our setting. To address this, we adopt unbalanced optimal transport, which allows for flexible mass redistribution between the two responses while preserving meaningful semantic differences.

The cornerstone of OTPO lies in constructing a cost matrix $M \in \mathbb{R}^{|y_c| \times |y_r|}$, where each entry M_{ij} quantifies the distance between the i-th token of the chosen response y_c and the j-th token of the rejected response y_r. Since the optimal transport framework requires the transported distribution to reside in a proper metric space, directly using the log-likelihood ratio difference in Eq. 6 is not feasible, as it does not naturally form a metric space. We take a step back and leverage the last-layer token representations, which better preserve the underlying semantic structure. We specifically use euclidean distance, i.e., $M_{ij} = \|h_c^i - h_r^j\|_2$, as it is commonly used in metric space (Arjovsky et al., 2017). Here, h_*^t is the hidden state of the t-th token in response y_*, extracted from the model's hidden representation space.

Building on this cost matrix, we define the optimization problem as:

$$\begin{aligned} \Gamma^* = \arg\min_{\Gamma} \sum_{i,j} \Gamma_{i,j} M_{i,j} + \epsilon_1 \Omega(\Gamma) \\ + \epsilon_2 (\mathbb{KL}(\Gamma \mathbf{1}, \mathbf{1}_{|y_c|}) + \mathbb{KL}(\Gamma^T \mathbf{1}, \mathbf{1}_{|y_r|})) \end{aligned} \tag{9}$$

where $\Gamma \in \mathbb{R}^{|y_c| \times |y_r|}$ represents the transport plan that aligns tokens between the chosen and rejected responses. $\Omega(\Gamma) = \sum_i \sum_j \Gamma_{i,j} \log(\Gamma_{i,j})$ is an entropy regularizer, controlling the sparsity of the transport plan. Meanwhile, the $\mathbb{KL}(\cdot)$ terms ensure that the marginal distributions of Γ are close to the naive DPO uniform weights, allowing for controlled deviations. This formulation unifies semantic alignment and token weight control under

an optimal transport framework, whereas the first term corresponds to the Wasserstein distance between the two responses' distribution.

After solving for the optimal transport plan Γ^*, the token-level weights ω_c^* and ω_r^* are obtained by summing along the respective dimensions as in Eq. 7 and normalizing to a predefined scale τ to ensure optimization stability:

$$\omega_c^* = \frac{\Gamma \mathbf{1}}{|\Gamma|}\tau, \quad \omega_r^* = \frac{\Gamma^\top \mathbf{1}}{|\Gamma|}\tau \tag{10}$$

Here, $|\Gamma|$ represents the total weight sum of the transport plan. The normalized transport plan is equal to the one in Eq. 9 when down-weighting the distance term by $\tau/|\Gamma|$. This allows for automatic total weight adjustment based on response length, therefore leading to a more stabilized total weight budget and preference optimization. We specifically set $\tau = \min(|y_c|, |y_r|)$ to ensure the total weight corresponds to the public length, which is enough to contain a more concise representation of relevant information most of the time. This choice helps balance the contributions of the two responses and reduces the disproportionate influence of less relevant tokens.

Then we incorporate the optimal token-level weights ω_c^* and ω_r^* in Eq. 10 into the reward estimation, replacing the uniform weights used in standard DPO. This allows the model to focus on semantically significant tokens, yielding a reward difference estimate:

$$\begin{aligned}\Delta_{\hat{r}} = &\sum_{i=1}^{|y_c|} \omega_c^{*i} \log \frac{\pi_\theta(y_c^i | x, y_c^{<i})}{\pi_{\text{ref}}(y_c^i | x, y_c^{<i})} \\ &- \sum_{j=1}^{|y_r|} \omega_r^{*j} \log \frac{\pi_\theta(y_r^j | x, y_r^{<j})}{\pi_{\text{ref}}(y_r^j | x, y_r^{<j})}\end{aligned} \tag{11}$$

This weighted reward difference captures the fine-grained contributions of individual tokens to the overall preference. The final OTPO loss is formulated as:

$$\mathcal{L}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_c,y_r)\sim D}\left[\log \sigma(\beta \Delta_{\hat{r}})\right] \tag{12}$$

In summary, OTPO leverages Optimal Transport to dynamically assign a fixed weight budget to token pairs based on their semantic relevance, enabling a fine-grained inspection of the reward difference term. This approach emphasizes meaningful token interactions while reducing the impact of less relevant or extraneous tokens, providing a more robust and interpretable optimization framework of LLM alignment.

3 Experimental Setup

We conduct preference optimization experiments to compare different optimization methods under various settings, including task, optimization strategy, and model. The tasks include general instruction-following and summarization. For general instruction-following, we conduct on-policy optimization on Llama-3-8B and Llama-3.2-3B, with UltraFeedback (Cui et al., 2024) and HelpSteer2 (Wang et al., 2025)'s preference version as the preference training dataset. For the summarization task, we perform offline optimization on Qwen-2.5-3B (Yang et al., 2024) using off-the-shelf TL;DR (Stiennon et al., 2020) dataset.

Model Training. In the general instruction-following task, we mainly consider off-the-shelf instruction-tuned models with more powerful instruction-following abilities. We first sample 10 responses per prompt for HelpSteer2 and 5 responses per prompt for UltraFeedback following (Meng et al., 2024; Tunstall et al., 2024; Wang et al., 2025). Then, we annotate the sampled responses with ArmoRM (Wang et al., 2024a) and select the response with the highest and lowest score as y_c, y_r respectively. In the summarization task, we first fine-tune `Qwen-2.5-3B` with the chosen responses, to obtain basic summarization capability, and then train the model directly with the TL;DR dataset. We tune general hyperparameters using DPO and apply the set of hyperparameters to most of all preference optimization methods. Please refer to Appx. A for more detailed descriptions.

Baselines. We primarily compare OTPO with DPO (Rafailov et al., 2024b) and other direct preference optimization methods, excluding RLHF approaches that require training an additional reward model, following prior works such as SimPO (Meng et al., 2024) and SamPO (Lu et al., 2024). Our focus is on methods that incorporate token-level weighting schemes. This includes SimPO, SamPO, and LDDPO (Liu et al., 2024b), which adjust the token weights to reduce total token weight differences and elevate length fairness. Please refer to Tab. 1 for more details and Fig. 3 for a more intuitive visualization of explicit token-level weighting schemes. We also include TDPO (Zeng et al., 2024), which implicitly applies token weighting via token-level KL divergence. Additionally, we also include other variants of DPO, including length regularized DPO (LR-DPO), AOT (Melnyk et al., 2024), Robust DPO (Ray Chowdhury et al.,

2024a), RSO (Liu et al., 2024a), SPPO (Wu et al., 2025) in Appx. C.

Evaluation. We assess the alignment performance using GPT to perform pairwise comparisons. We adopt AlpacaEval2 (Li et al., 2023; Dubois et al., 2024), to assess the models' instruction-following ability. AlpacaEval2 contains 805 questions, and uses GPT-4-Turbo to perform side-by-side comparisons of the model response with a reference model (GPT-4 Preview). We report the win rate, length-controlled win rate, and response average length. The length-controlled win rate is computed by first estimating the impact of length differences on each test result and then adjusting for a length difference of zero to obtain a debiased estimation. We specifically choose this benchmark, as it controls the effect of response length, while other LLM-as-a-judge benchmarks may exploit spurious correlations, including output length, presence of lists, position biases (Zheng et al., 2023; Koo et al., 2024; Wang et al., 2024c; Wu and Aji, 2025). For the summarization task, we follow (Rafailov et al., 2024b) and evaluate the win rate against the base model, using GPT-4o as the judge model on 256 randomly sampled test cases from the TL;DR test set. We also adopt MMLU, GSM8K, ARC Challenge, HellaSwag, and PiQA to examine the models' general ability on multiple domains in Appx. C.

4 Experimental Results

In this section, we present the main results of our experiments, demonstrating the superior performance of OTPO in various settings (Sec. 4.1). Then we conduct ablation studies to validate the components of OTPO (Sec. 4.2). Finally, we conduct a human evaluation for thorough evaluation (Sec. 4.3).

4.1 Main Results

We present the main results in Tab. 2 and Fig 4, showcasing OTPO's superiority in optimizing preferences across different backbones, tasks, and optimization strategies.

Overall preference enhancement of OTPO. As shown in Tab. 2, OTPO demonstrates the best result on length-controlled win rate across 2 backbones and 2 datasets, with an improvement of 2.6% to 10.9% increase compared to DPO, and 1.0% to 3.8% increase compared other baselines. Furthermore, it achieves the best win rate in 3 out of 4 settings, with up to 3.5% increase compared to the best baseline, demonstrating its robust effectiveness in improving alignment. Notably, OTPO is also better than other methods as shown in Tab. 9 in Appx. C.

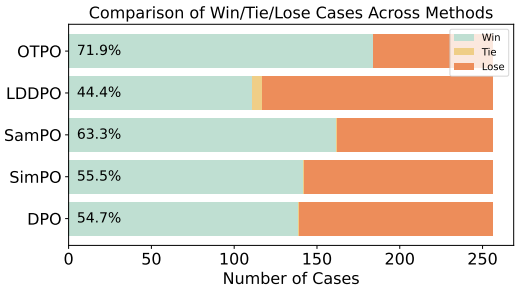


Figure 4: TL;DR summarization win rates compared to the base model, using GPT-4o as the evaluator. OTPO exceeds the existing methods by a large margin.

OTPO excels in the domain-specific task. In summarization task, OTPO exceeds other methods by a large margin of 8.6% win rate compared to the best-performing baseline SamPO, as shown in Fig. 4. As OTPO is trained to emphasize important parts in a response, it generates summarizations that are more concise and include key meanings. As the summarization task specifically requires concise summarizations, we regard all responses exceeding a certain length as "Lose" during evaluation following (Stiennon et al., 2020).

Mitigating Length bias. Although OTPO appears relatively less performant in the UltraFeedback setting with Llama-3-8B based on the naive win rate of 47.58%, this can be largely attributed to its production of substantially shorter responses, with an average length of 1791 tokens compared to the longer responses generated by other methods. Longer responses are not inherently problematic, especially if increased length leads to improved quality. However, (Zheng et al., 2023) has shown that "LLM judge favors longer, verbose responses, even if they are not as clear, high-quality, or accurate as shorter alternatives." Our goal with OTPO is not to shorten responses arbitrarily but to prioritize essential, high-quality content, with brevity emerging as a natural byproduct. By mitigating unnecessary verbosity, OTPO ensures that models focus on delivering information more concisely without sacrificing informativeness, as indicated by the highest length-controlled win rate 53.37%.

Length bias varies across settings. The results reveal opposing trends in length bias between the two datasets, particularly with Llama-3.2-3b-Instruct. In the UltraFeedback setting, there is a 3% increase in response length compared to the initial model, while in the HelpSteer2 setting, there is a

	Method	UltraFeedback			Helpsteer2		
		LC WR (%)	WR (%)	Length	LC WR (%)	WR (%)	Length
Llama-3-8b -Instruct	Initial	22.92	22.57	1899	22.92	22.57	1899
	DPO	48.14	<u>51.52</u>	2168	27.91	27.45	1945
	SimPO	47.56	40.72	1756	26.77	27.25	1984
	SamPO	<u>52.17</u>	46.31	1806	26.95	27.16	1991
	LDDPO	52.1	**51.72**	2036	<u>28.55</u>	<u>28.54</u>	1956
	OTPO	**53.37*****	47.58	1791	**29.64*****	**29.54**	1991
Llama-3.2-3b -Instruct	Initial	17.97	19.34	2041	17.97	19.34	2041
	DPO	26.02	<u>27.96</u>	2094	19.99	<u>20.54</u>	1970
	SimPO	22.58	22.34	1944	19.03	19.72	1996
	SamPO	24.08	26.49	2084	18.58	19.65	2018
	LDDPO	<u>26.56</u>	25.79	1909	<u>20.29</u>	20.2	1939
	OTPO	**26.97*****	**28.61**	2075	**20.5*****	**21.25**	2000

Table 2: AlpacaEval 2 evaluation results under four settings. WR denotes win rate, LC WR denotes length-controlled win rate. Models aligned using OTPO achieve superior performance on length-controlled win rates across all settings. The best results are marked in **bold**. The second-best results are <u>underlined</u>. Results marked with *** are significantly better than others with 99% confidence.

	LC WR	WR	Length
Initial	22.92	22.57	1899
DPO	48.14	<u>51.52</u>	2168
OTPO	**53.37**	47.58	1791
(1) Ablation of Optimal Transport			
Uniform	52.60	46.36	1796
Similarity	<u>53.28</u>	46.09	1757
(2) Ablation of Weight Normalization			
None	26.38	26.07	1939
Mean	52.79	46.69	1791
Max	49.85	44.77	1808
Length	48.51	**52.12**	2167

Table 3: Ablation study of OTPO on Llama-3-8B-Instruct with UltraFeedback. We ablate each component of OTPO: (1) (middle part) Replace OT weight with uniform weight or cosine similarity-based weight. (2) (lower part) Varying the weight sum normalization term τ by mean/max of the two responses' length, or each response length in Eq. 10 after OT.

3% decrease in response length. This suggests that length bias can manifest differently depending on the combination of the dataset and the optimized model, leading to overly lengthy or overly concise responses. We describe a more detailed analysis of generated on-policy datasets in Appx. B.

4.2 Ablation Study

We ablate each key design in OTPO: optimal transport guided weighting scheme, and weight normalization, and then report the results in Tab. 3.

4.2.1 Alternatives of Token Weighting Scheme

We replace the optimal transport-based token weighting scheme with uniform weight, or embedding similarity-based weight for comparison. Uniform weight only ensures response length fairness, while embedding similarity-based weight additionally applies a simplified algorithm with a similar motivation as OTPO.

Uniform weight ensures response length fairness. Uniform weight simply down-weight tokens in the longer response's weight to $\frac{\min(|y_c|,|y_r|)}{\max(|y_c|,|y_r|)}$ in DPO without OT ("Uniform"), so that each response's weight sum up to $\min(|y_c|, |y_r|)$. Compared to DPO, it decreases average response length (-372), thus trading off win rate (-10%) against length-controlled win rate (+9.3%).

Similarity-based weighting scheme improves performance. The embedding similarity-based weighting method ("Similarity") simplifies the OT process while adhering to the same intuition. For each token in the response, its representation is compared to the average representation of all tokens in the other response using cosine similarity. These similarities are then passed through a softmax function to calculate relative importance across all tokens in the response. Finally, we assign the same total weight budget $\min(|y_c|, |y_r|)$ based on the relative importance to obtain the final weight. This approach further improves length

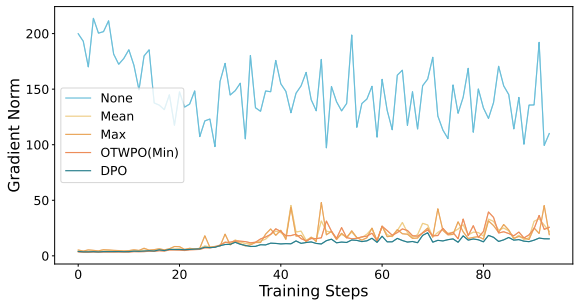


Figure 5: Trend of gradient norm during training.

fairness by considering token-level relationships between the chosen and rejected responses. However, it is slightly worse than OTPO in both length-controlled win rate and win rate, as it only considers the relationship of each token to the other response, while failing to consider more detailed token pair relationships.

4.2.2 Impact of Weight Normalization

We test various normalization strategies alongside OT, experimental indicates that they are worse than the original "min" normalization.

Other normalization variants lead to large fluctuations in training.. We further report the trend of gradient norm during training in Fig. 5. Using no normalization ("None") leads to significantly lower alignment performance, less than half of the best length-controlled win rate. This is mainly due to the large fluctuations in total weight $|\Gamma|$, thus leading to too aggressive gradient updates. Using mean or max for normalization leads to a decrease in both length-controlled win rate and win rate. This may relate to the large fluctuations in the shorter response's weight upscale across samples, which can lead to suboptimal performance.

Separating normalization for each response achieves the best win rate. We train a version where each response's weights are rescaled to match its original sum, i.e. $\tau = |y_c|$ for y_c and $\tau = |y_r|$ for y_r, alongside the application of OT ("Length"). This approach, while slightly improving both the length-controlled win rate and win rate over the DPO baseline, lags behind OTPO in terms of length-controlled win rate as it leads to a mismatch between the chosen and rejected responses' distributions, which affects overall alignment. These findings demonstrate the complementary roles of OT and weight normalization in optimizing both response quality and alignment.

	Expert1	Expert2
DPO	0.46	0.5
SimPO	<u>0.56</u>	<u>0.54</u>
SamPO	0.48	0.46
LDDPO	<u>0.56</u>	0.48
OTPO	**0.62**	**0.64**

Table 4: Human evaluation of the win rates of different methods compared to the base model. OTPO is considered the best by both experts.

4.3 Human Evaluation

We conduct human evaluations to further verify the effectiveness of OTPO. We randomly sample 50 questions across multiple domains as input, prompt the base model, i.e. `Llama-3-8B-Instruct`, and the model aligned using UltraFeedback to answer the question. Then we ask human experts to choose the better response based on relevance, coherence, completeness, and conciseness. The two responses' positions are randomly swapped to ensure evaluation fairness. See more details of human evaluation's setting in Appx. J.2. Tab. 4 reports win rates judged by human experts. We find the two experts' annotation results have a relatively low correlation of 0.37, which can be attributed to the diverse nature of human preference and the similarity across responses from the same model family. Results show that OTPO is considered the best among the two experts despite their diverse preferences.

5 Complexity and Efficiency Analysis

In this section, we analyze the computational and memory complexity of OTPO, mostly attributed to the optimal transport learning schema, and compare its efficiency with related methods.

Time Complexity. The optimal transport learning schema has a time complexity of $O(n^2)$, which is negligible compared to the transformer's forward pass complexity of $O(ln^2d + lnd^2)$, where n is the input length, d is the hidden dimension, and l is the model depth.

Memory Complexity. The OT step requires storing the pairwise cost matrix $M \in \mathbb{R}^{l\times l}$, two auxiliary vectors in $\mathbb{R}^l$, and an additional matrix of the same shape as M. This results in a memory complexity of $O(l^2)$, which is minor compared to the memory requirement of the transformer forward and backward passes, typically $O(l(n^2 + nd^2))$. As a result, OTPO introduces negligible additional memory overhead.

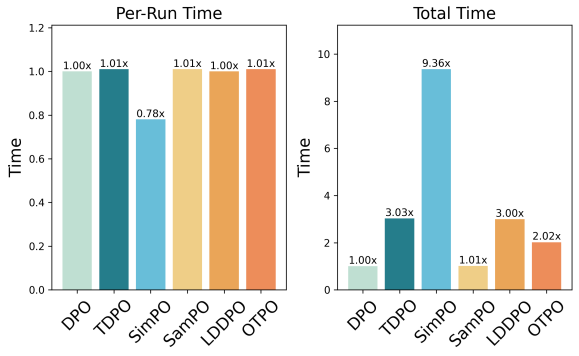


Figure 6: Comparison of training time across different preference optimization methods. We normalize DPO's training time as the baseline (1.00) and report each method's relative time as a fraction of it. *Per-Run Time* is for a single training run; *Total Time* includes time for hyperparameter tuning based on DPO's optimal settings.

Empirical Efficiency. As shown in Fig. 6, OTPO exhibits training efficiency comparable to existing preference optimization methods. Despite the slight increase in per-run cost due to the OT computation, OTPO remains favorable in terms of total training time, especially when hyperparameter tuning is considered. Given the optimal β for DPO, SimPO requires extensive tuning due to its sensitivity to its hyperparameters β, γ, while OTPO is more stable and thus requires fewer runs.

In summary, OTPO introduces minimal overhead in both time and memory while offering improved instruction-following performance. This favorable trade-off makes OTPO a practical and efficient choice for alignment training.

6 Related Work

Preference Optimization for LLMs. Preference optimization plays a vital role in aligning LLMs with human values and expectations. First proposed in (Ouyang et al., 2022), they train the policy using proximal policy optimization to maximize the estimated reward given by a trained reward model. Direct preference optimization eliminates the need for a reward model and directly leverages pairwise comparisons to guide models toward preferred behaviors. Many efforts have been taken to improve the offline learning objective or to reduce computational costs. RSO (Liu et al., 2024a), IPO (Gheshlaghi Azar et al., 2024), EXO (Ji et al., 2024), NCA (Chen et al., 2024), BCO (Jung et al., 2024), SPPO (Wu et al., 2025) replace the sigmoid function in DPO with other non-linear variants to model different preference objectives. CDPO (Mitchell, 2023), Robust DPO (R-DPO) (Ray Chowdhury et al., 2024b) enhances DPO by improving robustness to preference noise, and RPO (Pang et al., 2024) adds a negative log-likelihood term to prevent large decrease in chosen responses' probability. To tackle the commonly observed length bias, SimPO (Meng et al., 2024), SamPO (Lu et al., 2024), LDDPO (Liu et al., 2024b) apply a heuristical kind of token weighting scheme to elevate length fairness between two responses. APO (D'Oosterlinck et al., 2025) rewrites irrelevant parts with an external LLM to create minimally contrastive preference data. Our work builds upon these insights, introducing optimal transport to calculate token weights by assigning more importance to those semantically relevant tokens.

Optimal Transport for Machine Learning. Optimal Transport has proven to be a powerful tool in machine learning, particularly for tasks involving distribution alignment , such as transfer learning (Flamary et al., 2016; Courty et al., 2017), generative modeling, e.g. Wasserstein GANs (Arjovsky et al., 2017) and (Wang et al., 2024b), natural language processing (Asano et al., 2020), and recommendation (Han et al., 2024). In particular, recent efforts have also utilized OT to perform preference optimization on unpaired preference datasets by achieving distributional dominance (Melnyk et al., 2024). We apply OT to align token distributions between chosen and rejected responses in preference optimization, capturing the fine-grained differences in token semantics and context, and enabling a more principled weighting mechanism.

7 Conclusion

In this paper, we proposed an **O**ptimal **T**ransport-based token weighting scheme for direct **P**reference **O**ptimization (OTPO), a context-aware token weighting scheme to reinforce semantically meaningful differences in reward estimation. OTPO leverages optimal transport to dynamically assign a fixed total weight budget to each token pair in the chosen and rejected response based on their semantic similarity, and then aggregate each token pair's log-likelihood ratio difference as a contrastive reward difference estimate. It represents a step toward more robust and interpretable reward estimation and lays the groundwork for future exploration into fine-grained preference modeling in alignment tasks.

Limitations

Our experiments were limited to off-policy and on-policy setups for direct preference optimization. Recent research has highlighted that iterative on-policy setups may yield larger improvements in instruction-following performance. We did not explore such setups due to limited computational resources. This leaves room for future work to further enhance model performance using OTPO under iterative on-policy setups.

We conducted our experiments on relatively small models. While this scale provides meaningful insights, the scalability and generalizability of our algorithms to larger models, such as those with hundreds of billions of parameters, remain to be validated. Addressing this limitation would require significant computational resources and could further confirm the robustness of our approach across different model sizes.

For evaluating alignment quality, we relied on GPT-4 as the evaluation judge. While GPT-4 offers state-of-the-art evaluation capabilities, it may introduce potential biases and result in less accurate or reliable judgments. This could affect the evaluation of alignment improvements, and future work may explore more robust, unbiased, and possibly human-in-the-loop evaluation mechanisms.

Ethical Statements

We performed only a relatively simple check on the datasets used in our experiments. Although we made efforts to ensure the datasets were suitable for training alignment models, they may contain few harmful or inappropriate content. Addressing these issues requires more thorough dataset curation and filtering processes, which were beyond the scope of this work.

Due to the availability of datasets, our training experiments were conducted mainly on English datasets. We have not verified the generalizability or effectiveness of our algorithm on non-English datasets. This limitation underscores the need for future work to ensure alignment algorithms are robust across languages and culturally diverse contexts.

Our primary focus was on improving instruction-following abilities in aligned large language models. However, these models may still exhibit safety risks, such as generating harmful or biased outputs, which were not fully addressed in this study. Post-alignment safety evaluations and interventions are critical to mitigate such risks and ensure the responsible deployment of these models.

Acknowledgements

The authors would like to thank members of Shopee LLM Team for their helpful feedback and discussions, and the anonymous reviewers for their valuable suggestions. This work was partly supported by the National Natural Science Foundation of China (NSFC) (NO. 62476279, NO. U2436209), Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Renmin University of China, the Fundamental Research Funds for the Central Universities, and the Research Funds of Renmin University of China No. 24XNKJ18. This work was partially supported by fund for building world-class universities (disciplines) of Renmin University of China and Public Computing Cloud, Renmin University of China.

References

Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR.

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

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

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

Huayu Chen, Guande He, Lifan Yuan, Ganqu Cui, Hang Su, and Jun Zhu. 2024. Noise contrastive alignment of language models with explicit rewards. In *Advances in Neural Information Processing Systems*, volume 37, pages 117784–117812. Curran Associates, Inc.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias

Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. 2017. Joint distribution optimal transportation for domain adaptation. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, et al. 2024. Ultrafeedback: Boosting language models with scaled ai feedback. In *Forty-first International Conference on Machine Learning*.

Tri Dao. 2024. Flashattention-2: Faster attention with better parallelism and work partitioning. In *The Twelfth International Conference on Learning Representations*.

Karel D'Oosterlinck, Winnie Xu, Chris Develder, Thomas Demeester, Amanpreet Singh, Christopher Potts, Douwe Kiela, and Shikib Mehri. 2025. Anchored preference optimization and contrastive revisions: Addressing underspecification in alignment. *Transactions of the Association for Computational Linguistics*, 13:442–460.

Yann Dubois, Percy Liang, and Tatsunori Hashimoto. 2024. Length-controlled alpaca'eval: A simple debiasing of automatic evaluators. In *First Conference on Language Modeling*.

Rémi Flamary, Nicholas Courty, Davis Tuia, and Alain Rakotomamonjy. 2016. Optimal transport for domain adaptation. *IEEE Trans. Pattern Anal. Mach. Intell*, 1(1-40):2.

Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T.H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. 2021. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2024. A framework for few-shot language model evaluation.

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 *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 4447–4455. PMLR.

Zhongxuan Han, Chaochao Chen, Xiaolin Zheng, Meng Li, Weiming Liu, Binhui Yao, Yuyuan Li, and Jianwei Yin. 2024. Intra-and inter-group optimal transport for user-oriented fairness in recommender systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 8463–8471.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Haozhe Ji, Cheng Lu, Yilin Niu, Pei Ke, Hongning Wang, Jun Zhu, Jie Tang, and Minlie Huang. 2024. Towards efficient exact optimization of language model alignment. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 21648–21671. PMLR.

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

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. 2018. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR.

Ryan Koo, Minhwa Lee, Vipul Raheja, Jong Inn Park, Zae Myung Kim, and Dongyeop Kang. 2024. Benchmarking cognitive biases in large language models as evaluators. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 517–545, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

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

Tianqi Liu, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter J Liu, and Jialu Liu. 2024a. Statistical rejection sampling improves preference optimization. In *The Twelfth International Conference on Learning Representations*.

Wei Liu, Yang Bai, Chengcheng Han, Rongxiang Weng, Jun Xu, Xuezhi Cao, Jingang Wang, and Xunliang Cai. 2024b. Length desensitization in directed preference optimization. *arXiv preprint arXiv:2409.06411*.

Junru Lu, Jiazheng Li, Siyu An, Meng Zhao, Yulan He, Di Yin, and Xing Sun. 2024. Eliminating biased length reliance of direct preference optimization via down-sampled kl divergence. In *Proceedings of the*

2024 Conference on Empirical Methods in Natural Language Processing, pages 1047–1067.

Igor Melnyk, Youssef Mroueh, Brian Belgodere, Mattia Rigotti, Apoorva Nitsure, Mikhail Yurochkin, Kristjan Greenewald, Jiri Navratil, and Jarret Ross. 2024. Distributional preference alignment of LLMs via optimal transport. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

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

Eric Mitchell. 2023. A note on dpo with noisy preferences & relationship to ipo.

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. *Advances in neural information processing systems*, 35:27730–27744.

Richard Yuanzhe Pang, Weizhe Yuan, Kyunghyun Cho, He He, Sainbayar Sukhbaatar, and Jason Weston. 2024. Iterative reasoning preference optimization. In *Advances in Neural Information Processing Systems*, volume 37, pages 116617–116637. Curran Associates, Inc.

Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. 2024. Disentangling length from quality in direct preference optimization. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 4998–5017, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Rafael Rafailov, Joey Hejna, Ryan Park, and Chelsea Finn. 2024a. From r to q^*: Your language model is secretly a q-function. In *First Conference on Language Modeling*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024b. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3505–3506.

Sayak Ray Chowdhury, Anush Kini, and Nagarajan Natarajan. 2024a. Provably robust DPO: Aligning language models with noisy feedback. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 42258–42274. PMLR.

Sayak Ray Chowdhury, Anush Kini, and Nagarajan Natarajan. 2024b. Provably robust DPO: Aligning language models with noisy feedback. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 42258–42274. PMLR.

Jie Ren, Samyam Rajbhandari, Reza Yazdani Aminabadi, Olatunji Ruwase, Shuangyan Yang, Minjia Zhang, Dong Li, and Yuxiong He. 2021. Zero-offload: Democratizing billion-scale model training. In *2021 USENIX Annual Technical Conference (USENIX ATC 21)*, pages 551–564.

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*, volume 33, pages 3008–3021. Curran Associates, Inc.

Lewis Tunstall, Edward Emanuel Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro Von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M Rush, and Thomas Wolf. 2024. Zephyr: Direct distillation of LM alignment. In *First Conference on Language Modeling*.

Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. 2024a. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10582–10592, Miami, Florida, USA. Association for Computational Linguistics.

Jun Wang, Bohan Lei, Liya Ding, Xiaoyin Xu, Xianfeng Gu, and Min Zhang. 2024b. Autoencoder-based conditional optimal transport generative adversarial network for medical image generation. *Visual Informatics*, 8(1):15–25.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2024c. Large language models are not fair evaluators. In *Proceedings of the 61th Annual Meeting of the Association for Computational Linguistics (ACL 2023)*, pages 9440–9450.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Zhilin Wang, Alexander Bukharin, Olivier Delalleau, Daniel Egert, Gerald Shen, Jiaqi Zeng, Oleksii Kuchaiev, and Yi Dong. 2025. Helpsteer2-preference: Complementing ratings with preferences. In *The Thirteenth International Conference on Learning Representations*.

Minghao Wu and Alham Fikri Aji. 2025. Style over substance: Evaluation biases for large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 297–312, Abu Dhabi, UAE. Association for Computational Linguistics.

Yue Wu, Zhiqing Sun, Huizhuo Yuan, Kaixuan Ji, Yiming Yang, and Quanquan Gu. 2025. Self-play preference optimization for language model alignment. In *The Thirteenth International Conference on Learning Representations*.

Wenyi Xiao, Zechuan Wang, Leilei Gan, Shuai Zhao, Wanggui He, Luu Anh Tuan, Long Chen, Hao Jiang, Zhou Zhao, and Fei Wu. 2024. A comprehensive survey of datasets, theories, variants, and applications in direct preference optimization. *arXiv preprint arXiv:2410.15595*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024. Qwen2.5 technical report. *CoRR*, abs/2412.15115.

Jing Yao, Xiaoyuan Yi, Xiting Wang, Jindong Wang, and Xing Xie. 2023. From instructions to intrinsic human values–a survey of alignment goals for big models. *arXiv preprint arXiv:2308.12014*.

Xiaoyuan Yi, Jing Yao, Xiting Wang, and Xing Xie. 2023. Unpacking the ethical value alignment in big models. *arXiv preprint arXiv:2310.17551*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Yongcheng Zeng, Guoqing Liu, Weiyu Ma, Ning Yang, Haifeng Zhang, and Jun Wang. 2024. Token-level direct preference optimization. In *Forty-first International Conference on Machine Learning*.

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. *Advances in Neural Information Processing Systems*, 36:46595–46623.

A Implementation Details

We use UltraFeedback (Cui et al., 2024), the preference-transformed HelpSteer2 (Wang et al., 2025), and TL;DR (Stiennon et al., 2020) as preference training datasets . UltraFeedback contains 61,135 examples in its training split, and HelpSteer2 contains 9125 examples. TL;DR contains 92,534 summary comparisons for training, and 83,629 test comparisons. We generate the on-policy training data as follows: 1) Sample n responses for each prompt with the policy to be optimized. 2) Discard those samples where the n responses are identical. 3) Annotate the responses with a reward model, and select the response with the highest and lowest score as y_c, y_r. We set $n = 5$ for UltraFeedback, and $n = 10$ for HelpSteer2 following (Meng et al., 2024; Tunstall et al., 2024; Wang et al., 2025). The detailed statistics of the on-policy datasets are listed in Appx. B. Maximum prompt length and maximum input length are controlled as (2048, 1800) for the instruction-following task and (1024, 900) for the summarization task. This setting accommodates varying input sizes and prevents huge memory costs.

We first conduct preliminary hyperparameter search on learning rate in $\{3e^{-7}, 5e^{-7}, 7e^{-7}\}$, batch size in $\{64, 128, 256\}$, and epoch in $\{1, 2, 3\}$. Results show that training with a learning rate of $5e^{-7}$ and a batch size of 128 for 1 epoch typically yields the best results, so we use this set of hyperparameters for all experiments. Then we search β in $\{0.01, 0.05, 0.1\}$ for each setting using DPO, and set $\beta = 0.01$ for Llama-3-8B and Qwen-2.5-3B, $\beta = 0.1$ for Llama-3.2-3B across methods except for SimPO, which requires a much larger β. As for method-specific hyperparameters, we report the search ranges considered in Tab. 5. Optimization was performed using AdamW with a cosine schedule and warmup ratio of 0.1.

For the Optimal Transport part, each token's last-layer hidden state was used as its representation. We set $\epsilon_1 = 1$ for UltraFeedback and TL;DR, $\epsilon_1 = 0.1$ for HelpSteer2, and $\epsilon_2 = 0.2$ for all configurations. As for supervise fine-tuning Qwen-2.5-3B, we apply a batch size of 128 and a learning rate of $2e{-}5$ for 1 epoch.

We use the `alignment-handbook` library with Apache-2.0 license to perform preference optimization and supervise fine-tuning, incorporating the `POT` (Flamary et al., 2021) package to compute the optimal transport plan. For evaluation, we assess the model's instruction-following capabilities using the official `AlpacaEval2` repository. Additionally, we evaluate the model's general abilities using the Harness evaluation framework (Gao et al., 2024).

Method	Loss Function	Hyperparameter
DPO	$-\log\sigma(\beta\log\frac{\pi_\theta(y_c\|x)}{\pi_{\text{ref}}(y_c\|x)}-\beta\log\frac{\pi_\theta(y_r\|x)}{\pi_{\text{ref}}(y_r\|x)})$	$\beta\in[0.01,0.05,0.1]$
SimPO	$-\log\sigma(\frac{\beta}{\|y_c\|}\log\pi_\theta(y_c\|x)-\frac{\beta}{\|y_r\|}\log\pi_\theta(y_r\|x)-\gamma)$	$\beta\in[2.0,2.5,10,20]$ $\gamma\in[1,2,3]$
SamPO	$-\log\sigma(\beta\sum_{t=1}^{m}\log\frac{\pi_\theta(y_c^t\|x)}{\pi_{\text{ref}}(y_c^t\|x)}-\beta\sum_{t=1}^{m}\log\frac{\pi_\theta(y_r^m\|x)}{\pi_{\text{ref}}(y_r^m\|x)})$, where $m=min(\|y_c\|,\|y_r\|), y^t\sim\text{Uniform}(m,\{y\}^T)$	-
LDDPO	$-\log\sigma(\beta(\sum_{t=1}^{m}\log\frac{\pi_\theta(y_c^t\|x)}{\pi_{\text{ref}}(y_c^t\|x)}+\alpha\sum_{t=m+1}^{\|y_c\|}\log\frac{\pi_\theta(y_c^t\|x)}{\pi_{\text{ref}}(y_c^t\|x)})$ $-\beta(\sum_{t=1}^{m}\log\frac{\pi_\theta(y_r^t\|x)}{\pi_{\text{ref}}(y_r^t\|x)}+\alpha\sum_{t=m+1}^{\|y_r\|}\log\frac{\pi_\theta(y_r^t\|x)}{\pi_{\text{ref}}(y_r^t\|x)})), m=min(\|y_c\|,\|y_r\|)$	$\alpha\in[0.2,0.5,0.7]$
TDPO	$-\log\sigma((\beta\log\frac{\pi_\theta(y_c\|x)}{\pi_{\text{ref}}(y_c\|x)}-\beta\log\frac{\pi_\theta(y_r\|x)}{\pi_{\text{ref}}(y_r\|x)})$ $-\alpha(\beta D_{\text{SeqKL}}(x,y_r;\pi_{\text{ref}}\|\pi_\theta)-sg(\beta D_{\text{SeqKL}}(x,y_c;\pi_{\text{ref}}\|\pi_\theta))),$ where $D_{\text{SeqKL}}(x,y;\pi_{\text{ref}}\|\pi_\theta)=\sum_{t=1}^{\|y\|}D_{\text{KL}}(\pi_{\text{ref}}(\cdot\|[x,y^{<t}])\|\pi_\theta(\cdot\|[x,y^{<t}]))$	$\alpha\in[0.1,0.5,1.0]$
OTPO	$-\log\sigma(\beta(\sum_{i=1}^{\|y_c\|}\omega_c^{*i}\log\frac{\pi_\theta(y_c^i\|x,y_c^{<i})}{\pi_{\text{ref}}(y_c^i\|x,y_c^{<i})}-\sum_{j=1}^{\|y_r\|}\omega_r^{*j}\log\frac{\pi_\theta(y_r^j\|x,y_r^{<j})}{\pi_{\text{ref}}(y_r^j\|x,y_r^{<j})}))$	$\epsilon_1\in[0.1,1],\ \epsilon_2=0.2$

Table 5: Various preference optimization loss functions with weighting scheme and hyperparameters search range. Here, $\text{Uniform}(m,\{y\}^T)$ denotes uniformly sampling m tokens from all tokens in $\{y\}^T$, and sg represents the stop-gradient operator, which blocks the propagation of gradients.

	Llama-3-8B		Llama-3.2-3B	
	UF	HS	UF	HS
count	59876	9084	60692	9087
mean	-29	-24	-91	-75
std	233	168	447	257
min	-4058	-1854	-4094	-2016
25%	-67	-75	-99	-115
50%	-11	-14	-16	-30
75%	31	37	30	28
max	2218	1193	4079	1266

Table 6: Response length difference summary statistic of each on-policy dataset. UF denotes UltraFeedBack, while HS denotes HelpSteer2.

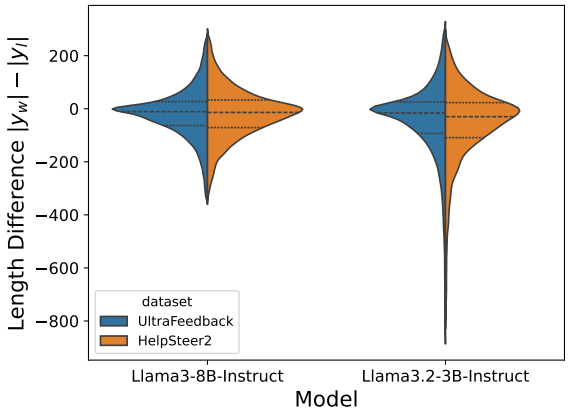


Figure 7: Dataset length difference ($|y_c| - |y_r|$) distribution. Dash lines indicated the quartiles.

As for the computation environment, we conducted training on 4xA100 for Llama-3-8B, and 2xA100 for Llama-3.2-3B, Qwen-2.5-3B using Pytorch. To accelerate training, we utilized FlashAttention2 (Dao, 2024), DeepSpeed Zero 3 (Rasley et al., 2020; Ren et al., 2021), and bfloat16 precision.

B On-policy Dataset Statistics

We present the statistics of length differences in the generated on-policy datasets in Tab. 6. To provide a clearer visualization of the length difference distribution, we leave out the top and bottom 2.5% of samples and visualize the remaining data using a violin plot in Fig. 7. The results show that prompts in HelpSteer2 tend to produce preference data pairs with more negative length differences, having a median of -30 compared to -16 for UltraFeedback with Llama-3.2-8B. Notably, the length differences in the long-tail regions are significantly smaller for Llama-3.2-3B, further highlighting the contrast between the two models.

We sample 50 samples from each on-policy dataset to verify that the data does not contain any information that uniquely identifies individual people of offensive content.

C General Ability Evaluation

C.1 Benchmark Description

To further validate the effect of OTPO on the models' general abilities, we evaluate the aligned models using 5 popular benchmarks:

	MMLU	GSM8K	ARC	HellaSwag	PiQA	Average
Llama-3-8b-Instruct + UltraFeedback						
Initial	65.64	75.59	56.57	57.70	78.18	66.74
DPO	65.80	74.98	56.83	56.07	74.92	65.72
SimPO	65.75	75.13	56.06	54.41	75.73	65.42
SamPO	65.62	70.43	55.46	53.98	74.21	63.94
LDDPO	65.73	71.65	57.17	55.13	74.70	64.87
OTPO	65.48	70.51	55.29	53.94	74.48	63.94
Llama-3-8b-Instruct + HelpSteer2						
Initial	65.64	75.59	56.57	57.70	78.18	66.74
DPO	65.79	75.97	56.66	57.87	78.73	67.00
SimPO	65.72	75.36	56.66	57.79	78.40	66.79
SamPO	65.81	75.66	56.91	57.91	78.40	66.94
LDDPO	65.69	75.13	56.74	57.93	78.40	66.78
OTPO	65.80	76.04	57.17	57.88	78.35	67.05
Llama-3.2-3b-Instruct + UltraFeedback						
Initial	59.71	64.14	45.90	52.34	75.63	59.54
DPO	60.06	66.49	46.84	52.48	75.84	60.34
SimPO	59.84	64.97	47.18	52.54	75.73	60.05
SamPO	59.72	65.88	46.84	52.53	76.01	60.20
LDDPO	60.00	66.03	47.18	52.49	75.79	60.30
OTPO	59.98	66.41	46.67	52.50	75.90	60.29
Llama-3.2-3b-Instruct + HelpSteer2						
Initial	59.71	64.14	45.90	52.34	75.63	59.54
DPO	59.64	64.29	46.16	52.38	75.68	59.63
SimPO	59.78	64.22	46.16	52.33	75.84	59.67
SamPO	59.68	63.53	46.08	52.22	75.90	59.48
LDDPO	59.67	63.99	46.16	52.37	75.63	59.56
OTPO	59.71	64.44	46.33	52.31	75.84	59.73

Table 7: General ability evaluation results across 4 settings.

- MMLU (Hendrycks et al., 2021): A massive multitask language understanding benchmark spanning broad domains. It consists of 4 subgroups and 57 subsets. We select the model's answer based on the probabilities of 'A', 'B', 'C', and 'D', as suggested in the original paper, and report the overall accuracy.

- GSM8K (Cobbe et al., 2021): A mathematical benchmark of grade school math problems for evaluating reasoning abilities. We evaluate the original test split with 1.32k examples and report the accuracy of answers with strict exact match.

- ARC Challenge (Clark et al., 2018): A more challenging benchmark involving complex reasoning over a diverse set of science exam questions, containing 2590 examples in the format of multiple choice. We report the normalized accuracy of overall test samples.

- HellaSwag (Zellers et al., 2019): A benchmark for predicting the endings of stories or scenarios, evaluating LLM's comprehension and creativity, targeted at commonsense reasoning via natural language. We evaluate the test split containing 10k examples and report the overall accuracy.

- PiQA (Bisk et al., 2020): A Physical Interaction Question Answering task to test physical commonsense reasoning, i.e. interaction with everyday objects in everyday situations. It contains 20k QA pairs that are either multiple-choice or true/false questions. We report the overall accuracy.

C.2 Evaluation Results

We report the general ability evaluation reports across 4 settings in Tab. 7

General ability assessment. OTPO exhibits comparable general ability to other baselines, with fluctuations less than 1% in most of the setups. It consistently achieves the highest average scores across both Llama-3-8B-Instruct (67.05) and Llama-3.2-3b-Instruct (59.73) in the HelpSteer2 setup. Moreover, it excels in reasoning ability, resulting in an increase in GSM8K (+0.15%) and ARC (+0.26%) compared to the best baseline. This highlights OTPO's potential to capture contextual and semantic differences to improve the model's reasoning ability.

Training dataset comparison. Helpsteer2 ensures consistent general ability improvement across most of the methods. This can be attributed to the difference in data distribution. While UltraFeedback mostly contains open-ended questions, HelpSteer2 contains more reasoning-related questions. However, training OTPO with UltraFeedback leads to a drastic decrease in Llama-3-8B-Instruct (-2.8), specifically on GSM8K(-5.08) and PiQA (-3.7). This phenomenon is commonly referred to as "alignment-tax", where models trade-off between general ability and instruction-following ability. This calls for more rigorous consideration when choosing the dataset for alignment according to specific alignment needs.

D Experiments on Other Models

We conducted additional experiments on Qwen-2.5-3B-Instruct and Mistral-7B-Instruct-v0.2 with on-policy datasets created from HelpSteer2 to further validate OTPO's effectiveness. The results are shown in Tab. 8

E Comparison to Other DPO Variants

OTPO remains the best algorithm compared to other DPO variants, along with the ones incorporating a certain kind of weighting scheme. We train Llama-3-8B on the HelpSteer2 dataset using these kinds of preference optimization loss:

- AOT (Melnyk et al., 2024) align LLMs by making the reward distribution of the chosen responses stochastically dominant in the first order on the distribution of rejected samples.

- BCO (Jung et al., 2024) trains a binary classifier that maps the chosen response to 1 and the rejected response to 0.

- CDPO (Mitchell, 2023) applies label smoothing to the original DPO loss, enhancing robustness to preference label noise.

- EXO (Ji et al., 2024) replaces the forward KL with reverse KL when deriving DPO loss.

- IPO (Gheshlaghi Azar et al., 2024) minimizes the squared loss of margin between the estimated reward margin and a predefined margin.

- NCA (Chen et al., 2024) optimizes the absolute likelihood of each response instead of the relative likelihood of two responses.

Method	Qwen-2.5-3B-Instruct			Mistral-7B-Instruct-v0.2		
	LC WR	WR	length	LC WR	WR	length
Initial	16.82	20.1	2145	17.74	15.05	1630
DPO	16.93	19.62	2100	27.25	24.78	1775
SimPO	16.75	19.47	2112	23.56	22	1906
SamPO	<u>18.26</u>	<u>20.75</u>	2105	27.31	25.31	1844
LDDPO	17.32	20.15	2107	<u>28.22</u>	<u>25.84</u>	1815
OTPO	**19.71*****	**22.06**	2074	**30.35*****	**28.2**	1839

Table 8: AlpacaEval 2 evaluation results with HelpSteer2 as the training dataset, Qwen-2.5-3B-Instruct and Mistral-7B-Instruct-v0.2 as the backbone model. WR denotes win rate, LC WR denotes length-controlled win rate. Models aligned using OTPO achieve the best performance on length-controlled win rates and win rates on both models. The best results are marked in **bold**. The second-best results are <u>underlined</u>. Results marked with *** are significantly better than others with 99% confidence.

	LC WR (%)	WR (%)	Length
Initial	22.92	22.57	1899
DPO	27.91	27.45	1945
AOT	28.55	27.83	1924
BCO	<u>29.23</u>	28.76	1955
CDPO	28.62	28.52	1954
EXO	27.22	26.82	1937
IPO	25.02	26.06	2021
NCA	27.72	27.82	1951
R-DPO	28.66	28.34	1940
RSO	28.93	<u>28.8</u>	1961
RPO	27.43	27.41	1963
SPPO	28.48	28.13	1946
LR-DPO	28.23	22.53	1654
SimPO	26.77	27.25	1984
SamPO	26.95	27.16	1991
LDDPO	28.55	28.54	1956
OTPO	**29.64**	**29.54**	1991

Table 9: A more comprehensive comparison of DPO variants on AlpacaEval2, with the lower part incorporating an implicit token weighting scheme. The best results are marked in **bold**. The second-best results are <u>underlined</u>.

- R-DPO (Ray Chowdhury et al., 2024b) model the probability of existing label noise and apply label smoothing.
- RSO (Liu et al., 2024a) replaces the sigmoid function with hinge loss.
- RPO (Pang et al., 2024) adds a negative log-likelihood loss of the chosen response to DPO loss, alleviating the decrease in the chosen reward.
- SPPO (Wu et al., 2025) treats the chosen and rejected response as two players, and solves the Nash equilibrium by pushing the chosen reward to 1/2 and the rejected reward to -1/2
- LR-DPO (Park et al., 2024) adds a length regularization term by adding a weighted length difference to the reward difference term.

We leave the incorporation of an optimal transport-based weighting scheme to these DPO variants for future work.

F Hyperparameter Sensitivity

We analyze the sensitivity of OTPO to hyperparameters ϵ_1, ϵ_2, focusing on their effects on absolute token weight difference, normalization value τ, and reward margin. Llama-3.2-3B and the HelpSteer2 dataset are utilized for this study. The results, presented in Fig. 10, reveal several key insights.

Impact of ϵ_1 on token weight difference. Higher values of ϵ_1 result in smaller weight differences, as the entropy regularization term imposes stronger penalization, leading to smoother weight distributions. On the other hand, ϵ_2 shows almost no impact on token weight difference.

Differing changes of ϵ_1, ϵ_2 on normalization value τ. A larger ϵ_2 decreases τ by imposing stronger penalties on marginal differences, whereas increasing ϵ_1 implicitly downweights the impact of marginal differences overall, leading to a higher τ. This interplay reflects the contrasting roles of these hyperparameters in shaping the weight normalization process, as well as the necessity of the normalization term τ in stabilizing training.

Stability of reward margin. Despite substantial variations in absolute token weight difference

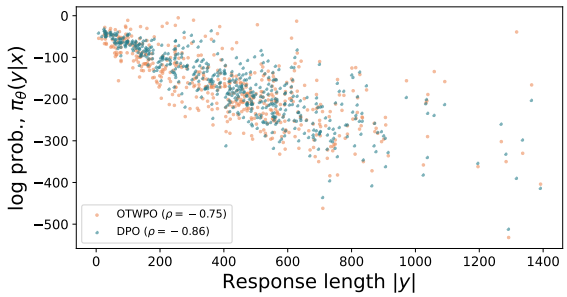


Figure 8: Estimated response log probability v.s. response length. OTPO exhibits a less positive trend in log probability with respect to DPO, leading to a more stable reward estimate.

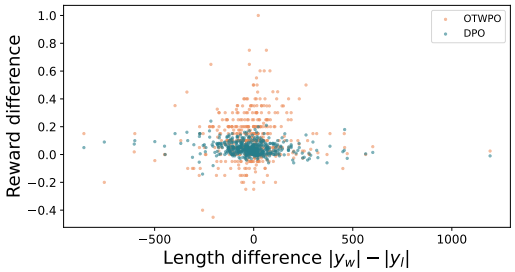


Figure 9: Reward difference v.s. the length difference between chosen and rejected responses. While the optimized reward differences by DPO are concentrated around 0.1, the reward differences optimized by OWTPO are more varied, especially larger when the length difference is low.

and τ, reward margins exhibit remarkable stability, The reward margin remains stable at approximately 0.12 across different configurations, and is consistently higher than DPO significantly. This stability can be attributed to the joint effects of Optimal Transport and weight sum normalization, which mitigate fluctuations in token weights while preserving the overall quality of reward estimation. Thus, OTPO ensures a stable reward estimation under varying hyperparameter configurations.

G Length Analysis

In this section, we further analyze OTPO's effect by comparing the sensitivity of response log probability and reward difference to response length (difference).

OTPO exhibits a relatively smaller positive relationship to length compared to DPO, as shown in Fig. 8. Specifically, the solid dots in the figure represent the actual test samples, while the dashed lines indicate linear fits based on these points. The gentler slope of OTPO relative to DPO suggests that OTPO provides more stable log probability estimates despite variations in response length. This aligns with the findings in Fig. 2, reinforcing OTPO's ability to mitigate length bias.

OTPO leads to more varied reward differences than DPO, attributing to its contextual awareness, as shown in Fig. 9. Unlike DPO, where reward differences remain narrowly distributed around 0.1 regardless of length differences, OTPO exhibits a broader range of reward differences, from -0.4 to 1.0. Notably, responses with larger absolute reward differences are associated with minimal length differences, indicating that OTPO's reward differences are driven primarily by contextual rather than length-related factors. This contextual sensitivity allows OTPO to dynamically optimize reward differences by focusing on meaningful content variations rather than superficial length differences. As a result, OTPO demonstrates superior adaptability and robustness in aligning reward optimization with instruction-following behavior.

H Case Study on the Transport Plan

We further analyze the transport plan Γ visualized as the Sankey diagram in Fig. 2, providing insights into the intuition behind optimal transport. In this diagram, each token is represented by a node, with tokens from the chosen response on the left and those from the rejected response on the right. Each node is labeled with its position in the pairwise data, its token text, and its aggregate weight ω_*^i. Token positions are indicated using response codes (C for chosen and R for rejected) and their indices, e.g., *C4* represents the 4th token in the chosen response. The lines connecting the nodes in the middle illustrate the transport flow, $\Gamma_{i,j}$, between token pairs, where the thickness of the lines reflects the magnitude of the flow. The height of each node is proportional to its aggregate weight, which corresponds to the sum of its inflow or outflow. To highlight the distribution of weights, we divide tokens into terciles: tokens in the highest tercile are colored teal, those in the middle tercile are light blue, and those in the lowest tercile are orange.

Shared and semantically similar tokens are densely mapped together. Tokens from the phrase "The capital of France is Paris" are assigned higher weights (> 1.5), as shown by the larger bars for these tokens and stronger connections between them. These parts directly address the question, making them critical for reward estimation. Conversely, less relevant tokens in the divergent parts, such as "known for its art" or "the birthplace of Vic-

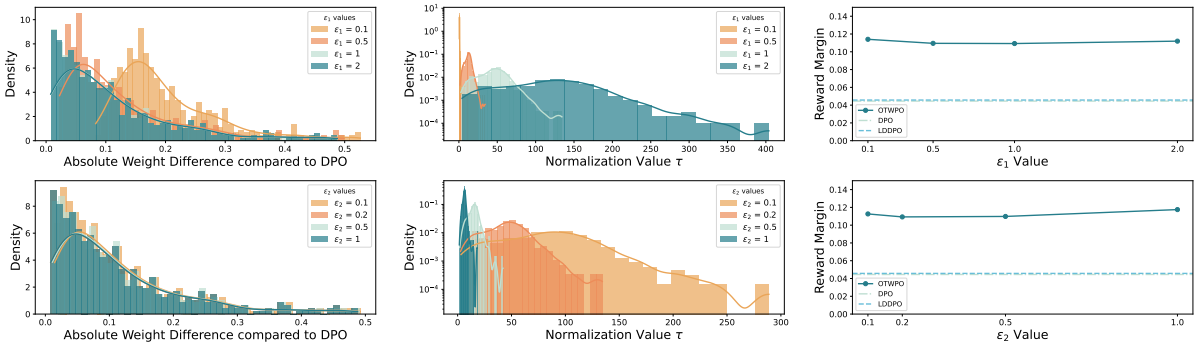


Figure 10: Hyperparameter sensitivity analysis of ϵ_1 (top), ϵ_2 (bottom). (left) Absolute token weight difference distribution between OTPO and DPO. (middle) The distribution normalization value τ in Eq.10. (right) Changes in reward margin.

tor Hugo" are assigned with lower weights (< 0.7), as they have thinner connections to other tokens and smaller bars. These parts, while potentially informative, are less central to the core instruction and are consequently de-emphasized. This weighting strategy ensures that the reward difference focuses on the most meaningful content, enhancing the stability of reward estimation.

A Smooth Transition of Weight. The Sankey diagram also illustrates a smooth transition of token weights, as optimal transport gradually shifts the emphasis from shared, contextually important tokens to less relevant, unique tokens. Despite the large weights (> 1.5) received by the upper shared part, other contextually important tokens receive moderately high weights ($0.8 - 0.9$), including the token "city" describing Paris, and the period marking the end of a sentence. This progression is evident in the intermediate-sized bars and connections associated with these tokens. As the responses diverge semantically, the weights of unique tokens progressively diminish (< 0.7). Tokens like "known for its art" or "the birthplace of Victor Hugo" cannot be well-mapped to corresponding tokens in the other response and therefore spread out thinly across many connections, as shown by the sparse and diffused lines in the lower part of the figure. This smooth transition ensures a natural weighting scheme that reflects the semantic relevance of each token.

I Connection to Existing Rewards

We use leave-one-out to explain the importance of each token to the final reward signal produced by current reward models, and compare the generated explanation with OT weight.

Generating token-level explanations. To compare the weighting scheme with existing rewards, we generate token-level explanations of explicit or implicit rewards. We applied a naive leave-one-out method, where each token was iteratively replaced with a padding token, and the resulting change in the reward score was measured. The proportional reduction in the reward score was treated as the token's contribution.

Method	Weight Visualization
OT Weight	The capital of France is Paris , a major European city known for its art , culture , and history .
ArmoRM	The capital of France is Paris , a major European city known for its art , culture , and history .

Table 10: Token weight visualization, darker color denotes higher weight. OT represents weights computed via Optimal Transport, while ArmoRM, DPO, and OTPO denotesthe naive leave-one-out explanations of reward prediction.

Connection to explicit reward model. We observed a high correlation of 0.76 between token-level weights computed using Optimal Transport and the explanation of predictions by an external reward model, ArmoRM (Wang et al., 2024a), as shown in Tab. 10. ArmoRM was selected for its compatibility with the tokenizer of the backbone model, Llama-3-8B, ensuring consistent tokenization. Both approaches focus strongly on the straightforward response to the prompt while downweighting more detailed explanations. Interestingly, since the rejected response contains a similar expression of the straightforward answer *"The capital of France is Paris"*, the OT weight-

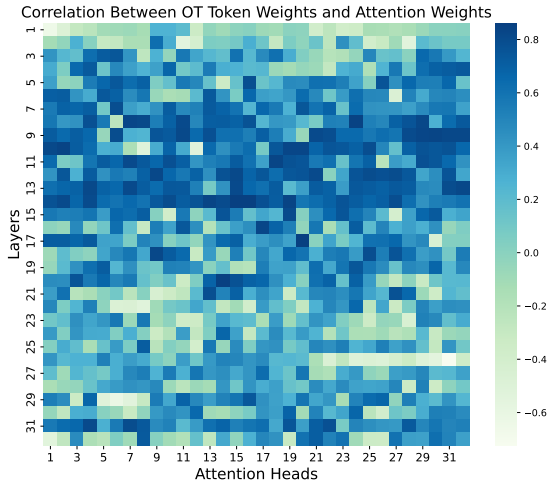


Figure 11: Spearman correlation between token weighting scheme by Optimal Transport and the last token attention weights of ArmoRM at each layer and attention head.

ing scheme also emphasizes this part. This aligns with the intuition behind OTPO, which prioritizes shared components that are more likely to be relevant to the prompt.

Relationship between OT weights and attention weights. We further examine the correlation between token-level weights derived from Optimal Transport and the attention weights of ArmoRM, as illustrated in Fig. 11. The highest correlations are predominantly found in the middle layers, where the model aggregates sequential context and refines its understanding of the input. This observation aligns with prior findings that model explanations often rely heavily on the middle layers (Kim et al., 2018), where meaningful internal representations emerge. Interestingly, despite lower correlations between OT weights and attention weights in the upper and lower layers, the leave-one-out explanations in Tab. 10 still exhibit strong overall agreement with OT. This implies that the middle layers are critical in bridging the gap between token-level contributions and global model explanations.

J Evaluation Configuration Details

J.1 Summarization Win Rate Calculation

In this section, we include the details for GPT-4o to generate win rates for summarization. The order of summaries is randomly chosen for every evaluation. Considering the special characteristic of **short** for summaries, we prompt the model to generate less than 48 words, and treat the summaries with more than 48 words as "Lose". Below is the prompt for GPT-4o.

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

Post:
{post}

Summary A:
{summary_A}

Summary B:
{summary_B}

FIRST provide a one-sentence comparison
of the two summaries, explaining which
you prefer and why. SECOND, on a new
line, state only "A" or "B" to indi-
-cate your choice.

Your response should use the format:
Comparison: <one-sentence comparison
and explanation>
Preferred: <"A" or "B">
```

J.2 Human Evaluation

In our human evaluation, we recruited human experts to choose the better response among two responses to the same question according to certain requirements. The human experts possess a high school level of English proficiency, allowing for easy comprehension of the responses. These experts were selected from within our academic institution to ensure a consistent educational background. To maintain the quality of annotation, we implemented a compensation structure that rewards the experts based on the number of pairwise responses they annotate. This approach was designed to incentivize thorough and careful consideration of each response pair.

During the evaluation, the experts were required to complete their assessments within a two-minute window for each response. This time constraint was established to simulate a realistic scenario in which users need to make quick judgments about the preference of responses. Both of the experts were presented with the same set of 50 pairwise responses for each method, totaling 250 pairwise

responses, to ensure consistency in the evaluation process.

Given the two responses to the same question, the evaluator needs to compare them from five perspectives, including relevance, coherence, completeness, conciseness, and instruction following. Then, the evaluator should make a judgment on which is the better response (winning response), or if the two responses are equally good/bad. The winning response will receive a score of 1, while the losing response will receive a score of 0. When the two responses are considered equally good/bad, then they both receive a score of 0.5.

We show guidelines that were provided to the evaluators in Fig. 12. These guidelines were crafted to assist the evaluators in their task and to standardize the evaluation criteria.

Human Evaluation Guideline

1. Evaluation Format

Each evaluation task presents a **prompt** and two **responses** (Response A and Response B). Evaluators need to compare the two responses based on evaluation criteria listed below and select the better response. If the responses are equally good or bad, evaluators can choose "Tie."

Example format:

Prompt: [Displayed]

Response A: [Displayed]

Response B: [Displayed]

Your Choice: A(Response A)/**B**(Response B)/**T**(Tie)

Optional Comments: [Free-text box]

2. Evaluation Criteria

Evaluators should compare the responses based on the following aspects:

Relevance & Accuracy: Does the response correctly address the prompt? Is the information factually accurate and relevant?

Coherence & Fluency: Is the response well-structured and grammatically correct? Does it read naturally and make logical sense?

Completeness: Does the response provide enough information to fully answer the prompt? Does it leave out key details?

Conciseness: Is the response clear and to the point without unnecessary verbosity? Does it avoid redundancy?

Instruction Following: If the prompt contains specific instructions, does the response adhere to them? Does it ignore or misinterpret any constraints or requirements?

3. Evaluation Options

For each evaluation task, evaluators must select one of the following options:

- **Response A is better** (A outperforms B in most criteria)
- **Response B is better** (B outperforms A in most criteria)
- **Tie** (Both responses are equally good or equally bad)

4. Examples

Example 1 (Clear Difference in Relevance & Accuracy)

Prompt: What is the capital of France?

Response A: Paris.

Response B: Berlin.

Correct Choice: Response A is better (B is factually incorrect).

Example 2 (Tie due to Equal Performance)

Prompt: Write a short poem about the ocean.

Response A: "The waves dance under the moon, a melody soft and bright."

Response B: "Beneath the sun, the ocean sings, a tune so vast and deep."

Correct Choice: Tie (Both responses are creative and valid).

Example 3 (Coherence & Fluency Issue)

Prompt: Summarize the importance of photosynthesis.

Response A: "Photosynthesis is the process by which plants convert sunlight into energy, producing oxygen as a byproduct."

Response B: "Plant make food sun energy. Oxygen too."

Correct Choice: Response A is better (B lacks coherence and fluency).

5. Payment

Evaluators will be compensated $0.25 per completed evaluation. Payment is based on task completion and quality control measures to ensure reliable judgments. Evaluators with consistently low-quality judgments may be disqualified from further participation.

Thank you for contributing to this evaluation! Your judgments help improve AI model performance and reliability.

Figure 12: Human Evaluation Guideline.

K Mathematical Derivations

K.1 Token-level DPO loss

We can transform a response y's probability $\pi(y|x)$ given x as follows:

$$\pi(y|x) = \prod_{i=1}^{|y|} \pi(y^i|y^{<i}, x) = \exp(\log \prod_{i=1}^{|y|} \pi(y^i|y^{<i}, x)) = \exp(\sum_{i=1}^{|y|} \log \pi(y^i|y^{<i}, x)) \tag{13}$$

For the reward difference term in Eq. 1:

$$\Delta_r = \log \frac{\pi_\theta(y_c|x)}{\pi_{\text{ref}}(y_c|x)} - \log \frac{\pi_\theta(y_r|x)}{\pi_{\text{ref}}(y_r|x)} \tag{14}$$

Since the likelihood of a response is modeled as the multiplicative probability of each token:

$$\pi_*(y|x) = \prod_{i=1}^{|y|} \pi_*(y^i|y^{<i}, x) \tag{15}$$

We can express Δ_r as:

$$\Delta_r = \log \frac{\prod_{i=1}^{|y_c|} \pi_\theta(y_c^i|y_c^{<i}, x)}{\prod_{i=1}^{|y_c|} \pi_{\text{ref}}(y_c^i|y_c^{<i}, x)} - \log \frac{\prod_{j=1}^{|y_r|} \pi_\theta(y_r^j|y_r^{<j}, x)}{\prod_{j=1}^{|y_r|} \pi_{\text{ref}}(y_r^j|y_r^{<j}, x)} \tag{16}$$

$$= \sum_{i=1}^{|y_c|} \log \frac{\pi_\theta(y_c^i|y_c^{<i}, x)}{\pi_{\text{ref}}(y_c^i|y_c^{<i}, x)} - \sum_{j=1}^{|y_r|} \log \frac{\pi_\theta(y_r^j|y_r^{<j}, x)}{\pi_{\text{ref}}(y_r^j|y_r^{<j}, x)} \tag{17}$$

K.2 Token Weight Derivation

We provide the derivations of the results in Tab. 1.

SimPO. The loss in (Meng et al., 2024) can be transformed as:

$$\mathcal{L}_{simpo} = -\log \sigma \left(\frac{\beta}{|y_c|} \log \pi_\theta(y_c|x) - \frac{\beta}{|y_r|} \log \pi_\theta(y_r|x) - \gamma \right) \tag{18}$$

$$= -\log \sigma \left(\frac{\beta}{|y_c|} \sum_{i=1}^{|y_c|} \log \pi(y_c^i|y_c^{<i}, x)) - \frac{\beta}{|y_r|} \sum_{j=1}^{|y_r|} \log \pi(y_r^j|y_r^{<j}, x) - \gamma \right) \tag{19}$$

$$= -\log \sigma \left(\beta \sum_{i=1}^{|y_c|} \frac{1}{|y_c|} \log \pi(y_c^i|y_c^{<i}, x)) - \beta \sum_{j=1}^{|y_r|} \frac{1}{|y_r|} \log \pi(y_r^j|y_r^{<j}, x) - \gamma \right) \tag{20}$$

Here, γ controls the overall margin to be optimized.

SamPO. Assume the shorter response's length is $m = min(|y_c|, |y_r|)$, the loss in (Lu et al., 2024) can be transformed as:

$$\mathcal{L}_{SamPO} = -\log \sigma \left(\beta \sum_{t=1}^{m} \log \frac{\pi_\theta(y_c^t|x)}{\pi_{\text{ref}}(y_c^t|x)} - \beta \sum_{t=1}^{m} \log \frac{\pi_\theta(y_r^m|x)}{\pi_{\text{ref}}(y_r^m|x)} \right), \text{ where } y^t \sim \text{Uniform}(m, y^T) \tag{21}$$

Here Uniform(m, y^T) denotes uniformly sample m tokens from all tokens in response y. Moving the sampling operation to the token index, we have:

$$\mathcal{L}_{SamPO} = -\log \sigma \left(\beta \sum_{t \in S_c} \log \frac{\pi_\theta(y_c^t|x, y_c^{<t})}{\pi_{\text{ref}}(y_c^t|x, y_c^{<t})} - \beta \sum_{t \in S_r} \log \frac{\pi_\theta(y_r^t|x, y_r^{<t})}{\pi_{\text{ref}}(y_r^t|x, y_r^{<t})} \right), \tag{22}$$

$$\text{where } S_* \sim \text{Uniform}(m, [1, |y_*|]) \tag{23}$$

Here Uniform$(m, [1, |y_*|])$ denotes uniformly sample m numbers from all integers from 1 to $|y_*|$.

LDDPO. (Liu et al., 2024b) transforms the probability of response to:

$$\hat{\pi}(y|x) = \prod_{i=1}^{m} \pi(y^i|x, y^{<i}) \prod_{i=m+1}^{|y|} \pi^{\alpha}(y^i|x, y^{<i}) \tag{24}$$

$$= \exp(\log \prod_{i=1}^{m} \pi(y^i|x, y^{<i}) \prod_{i=m+1}^{|y|} \pi^{\alpha}(y^i|x, y^{<i})) \tag{25}$$

$$= \exp(\sum_{i=1}^{m} \log \pi(y^i|y^{<i}, x) + \sum_{i=m+1}^{|y|} \alpha \log \pi(y^i|y^{<i}, x)) \tag{26}$$

Thus the loss can be derived as:

$$\mathcal{L}_{LDDPO} = -\log \sigma \left(\beta \log \frac{\hat{\pi}_\theta(y_c|x)}{\hat{\pi}_{\text{ref}}(y_c|x)} - \beta \log \frac{\hat{\pi}_\theta(y_r|x)}{\hat{\pi}_{\text{ref}}(y_r|x)} \right) \tag{27}$$

$$\begin{aligned} = -\log \sigma (\beta (\sum_{t=1}^{m} \log \frac{\pi_\theta(y_c^t|x)}{\pi_{\text{ref}}(y_c^t|x)} + \alpha \sum_{t=m+1}^{|y_c|} \log \frac{\pi_\theta(y_c^t|x)}{\pi_{\text{ref}}(y_c^t|x)}) \\ - \beta (\sum_{t=1}^{m} \log \frac{\pi_\theta(y_r^t|x)}{\pi_{\text{ref}}(y_r^t|x)} + \alpha \sum_{t=m+1}^{|y_r|} \log \frac{\pi_\theta(y_r^t|x)}{\pi_{\text{ref}}(y_r^t|x)})), \ m = min(|y_c|, |y_r|) \end{aligned} \tag{28}$$

K.3 Gradient Analysis

The derivative for the log-sigmoid function is:

$$\frac{\delta \log \sigma(u)}{\delta u} = \frac{1}{\sigma(u)} \frac{\delta \sigma(u)}{\delta u} = \frac{1}{\sigma(u)} (\sigma(u)(1 - \sigma(u))) = (1 - \sigma(u)) = \sigma(-u) \tag{29}$$

We can derive the gradient of OTPO as:

$$\nabla \mathcal{L}_{\text{OTPO}}(\pi_\theta) = -\beta \mathbb{E}_D \sigma(-\beta \Delta_{\hat{r}}) \nabla(\Delta_{\hat{r}}) \tag{30}$$

$$\Delta_{\hat{r}} = \sum_{i=1}^{|y_c|} \omega_c^{*i} \log \frac{\pi_\theta(y_c^i|x, y_c^{<i})}{\pi_{\text{ref}}(y_c^i|x, y_c^{<i})} - \sum_{i=1}^{|y_r|} \omega_r^{*i} \log \frac{\pi_\theta(y_r^i|x, y_r^{<i})}{\pi_{\text{ref}}(y_r^i|x, y_r^{<i})} \tag{31}$$

$$\nabla(\Delta_{\hat{r}}) = \sum_{i=1}^{|y_c|} \omega_c^{*i} \nabla_\theta \log \pi_\theta(y_c^i|x, y_c^{<i}) - \sum_{i=1}^{|y_r|} \omega_r^{*i} \nabla_\theta \log \pi_\theta(y_r^i|x, y_r^{<i}) \tag{32}$$

Similarly, the gradient of DPO is:

$$\nabla \mathcal{L}_{\text{DPO}}(\pi_\theta) = -\beta \mathbb{E}_D \sigma(-\beta \Delta_r) \nabla(\Delta_r) \tag{33}$$

$$\Delta_r = \sum_{i=1}^{|y_c|} \log \frac{\pi_\theta(y_c^i|x, y_c^{<i})}{\pi_{\text{ref}}(y_c^i|x, y_c^{<i})} - \sum_{i=1}^{|y_r|} \log \frac{\pi_\theta(y_r^i|x, y_r^{<i})}{\pi_{\text{ref}}(y_r^i|x, y_r^{<i})} \tag{34}$$

$$\nabla(\Delta_r) = \sum_{i=1}^{|y_c|} \nabla_\theta \log \pi_\theta(y_c^i|x, y_c^{<i}) - \sum_{i=1}^{|y_r|} \nabla_\theta \log \pi_\theta(y_r^i|x, y_r^{<i}) \tag{35}$$

We compare the gradients of OTPO and DPO to understand the impact on training and alignment results. For the reward difference term in Eq. 31, 34, OTPO is relatively smaller and more stable, as those semantically dissimilar tokens are down-weighted. While for gradient updates in Eq. 32, 35, the gradient scale of OTPO on each token is additionally controlled by the OT weighting scheme instead of uniform update, performing larger updates on the more important tokens related to prompt and the other response, while downweighting the updates in the less relevant tokens. This ensures a more meaningful and concentrated gradient update compared to DPO. Overall, the OT weighting scheme ensures more stable reward difference term and dynamic gradient updates given the context and pairwise data information.