PLOT: Enhancing Preference Learning via Optimal Transport

Anonymous ACL submission

Abstract

Preference learning in large language models (LLMs) has primarily followed two approaches: (1) fine-tuning-based methods that optimize models using human preference signals and (2) inference-phase techniques that regulate outputs through decoding-time interventions. While these methods effectively mitigate harmful content generation, they remain vulnerable to adversarial jailbreak attacks and suffer from limitations such as high computational costs, sensitivity to hyperparameters, and insufficient consideration of global token-level relationships. This paper introduces **PLOT**, a method that enhances the **P**reference **L**earning capability of fine-tuning-based alignment techniques through a token-level loss term derived from **O**ptimal **T**ransport. By modeling preference learning as an **Optimal Transport Problem**, PLOT aligns model outputs with human preferences while preserving the model's original distribution, thereby ensuring stability and robustness. Additionally, PLOT incorporates **token embeddings** to capture rich semantic relationships, enabling a more globally informed optimization process. Our experimental evaluations demonstrate that PLOT **significantly reduces attack success rates (ASR) across various red-teaming adversarial attacks** while maintaining general model performance. Compared to baseline fine-tuning methods, PLOT achieves **a reduction of up to 8.83% in ASR** while preserving fluency and coherence in general tasks. These results establish optimal transport as a principled and effective approach to preference learning, offering a robust framework for enhancing model alignment, safety, and adversarial robustness.

1 Introduction

Large language models (LLMs) have demonstrated increasingly remarkable capabilities and are being increasingly used to construct powerful AI systems that have a growing, sustained impact on the development of humanity (Kaplan et al., 2020; Bubeck et al., 2023; Brown et al., 2020; Achiam et al., 2023; Wei et al., 2021; Dubey et al., 2024; Liu et al., 2024a; Guo et al., 2025). In this context, human alignment for LLMs is crucial, as it ensures that models learn human preferences, producing outputs that are safe, reliable, and suitable for real-world applications (Christian, 2021; Gabriel, 2020; Kenton et al., 2021). This process is progressive, beginning with the need for safety and reliability as prerequisites for widespread accessibility (Gabriel and Ghazavi, 2021). Given the vast and sometimes overly comprehensive capabilities acquired during pre-training, controlling model outputs during real-world usage is essential (Ziegler et al., 2019; Ouyang et al., 2022; Peng et al., 2023). Numerous methods have been proposed to control the generation of safe content in model outputs. Concurrently, various jailbreak methods have emerged to assess and challenge the core safety aspects of model behavior, exploiting model vulnerabilities to evaluate their ability to generate harmless content (Deng et al., 2023; Shen et al., 2024; Yi et al., 2024). These developments collectively contribute to advancing safer and more reliable AI systems.

After meeting basic safety requirements, efforts have also been directed toward aligning models with a broader range of human preferences. Due to the substantial variability in human subjectivity, this alignment is currently constrained to more general conditions, such as the length of generated content (Gu et al., 2024), text quality (Stiennon et al., 2020), and the prevalence of hallucinations (Perković et al., 2024). In the domain of code generation, the focus is on producing correct executable code and addressing issues such as time and space complexity (Xu et al., 2022; Zhuo et al., 2024; Yang et al., 2024). Collectively, these aspects can be referred to as human preferences, and the process by which models learn to align with these preferences is termed preference learning, or alternatively, preference modeling.

While inference-phase alignment methods regulate outputs during decoding, fine-tuning-based approaches allow the model to internalize preferences, leading to more stable performance and a closer match to human behavior during skill acquisition. Building upon the superficial alignment hypothesis (Zhou et al., 2023), some works have attempted to enhance the quality of fine-tuning data to activate a preference subdistribution, thereby achieving improved performance (Chen et al., 2023; Liu et al., 2024b). Other approaches have focused on the distributional perspective, designing different loss functions based on token positions or probabilities in the output distribution to achieve preference learning (Zheng et al., 2023; Qi et al., 2024; Zhu et al., 2024). However, these approaches primarily focus on specific positions or independently consider individual tokens, without taking into account global information or the semantic relationships between tokens. As a result, they face several key challenges:

- *High computational cost*: Many existing methods require substantial computational resources due to complex loss functions and optimization constraints.

- *Limited performance improvements*: Current approaches often rely on localized token modifications rather than optimizing preferences holistically across the output distribution.

- *Sensitivity to hyperparameters*: Fine-tuning-based alignment methods are highly dependent on hyperparameter selection, limiting their robustness across diverse datasets and preference tasks.

To address these limitations, we propose a general, stable preference learning loss that integrates Optimal Transport (Villani et al., 2009) to enhance the preference learning capacity of existing fine-tuning alignment methods. Specifically, we compute the output distribution of the model and select a distribution that represents the target preference. By using optimal transport to calculate the minimal transportation distance between these distributions, we can stably measure the preference difference while preserving the model's original distribution (Arjovsky et al., 2017). Additionally, this approach incorporates the embedding of each token, capturing semantic information within the embedding space. Experimental results demonstrate that this loss significantly improves the preference learning performance of the original fine-tuning methods without compromising the model's general capabilities.

In summary, the contributions of this work are as follows:

1. We reformulate token-level preference learning as an optimal transport problem, introducing a novel preference learning loss that incorporates the semantic structure of the output distribution, providing a new perspective for model alignment.

2. Through extensive experiments, we demonstrate that integrating the proposed loss function into fine-tuning methods leads to better preference learning without compromising the model's general capabilities.

2 Related Works

2.1 Human Alignment

One of the most established approaches to human alignment in LLMs is Reinforcement Learning with Human Feedback (RLHF) (Bai et al., 2022a; Christiano et al., 2017), which is typically implemented with Proximal Policy Optimization (PPO) (Schulman et al., 2017). Although RLHF has been widely adopted in the training of advanced LLMs such as GPT-4 and Claude, it presents several challenges, including reward hacking, high sample complexity, and training instability, which limit its effectiveness in large-scale real-world applications.

Current alignment methods can be categorized into two primary approaches: fine-tuning alignment methods, which adjust model parameters, and inference-phase alignment methods, which constrain outputs without modifying parameters.

Fine-tuning alignment methods aim to internalize human preferences within the model's learned representations. These approaches can be further divided into:

- Reinforcement learning-based (RL-based) optimization: This category includes methods that integrate reinforcement learning techniques to optimize reward-based alignment strategies. Notable examples include Constitutional AI (Bai et al., 2022b), reinforcement learning with adversarially filtered datasets (Lee et al., 2023), and other self-improving fine-tuning methods that leverage large-scale

preference signals (Hu et al., 2023; Li et al., 2023c; Shao et al., 2024). While reinforcement learning provides a powerful mechanism for alignment, it is computationally expensive, sensitive to reward model design, and prone to instability.

- Fine-tuning-only approaches: Unlike RLHF, these methods rely entirely on supervised preference learning without the need for RL-based reward optimization. Fine-tuning-only strategies have been explored in models such as SteerLM (Dong et al., 2023), Ranked Reward Hyperparameter-Free Fine-tuning (RRHF) (Yuan et al., 2023), and Statistical Preference Optimization (SPO) (Liu et al., 2023). More recently, Direct Preference Optimization (DPO) (Rafailov et al., 2024) and its extensions (Morimura et al., 2024; Singhal et al., 2024; Pal et al., 2024) have been proposed to optimize LLMs directly from preference rankings without the complexity of reward models. These fine-tuning-only approaches generally offer greater stability and efficiency compared to RL-based optimization but may require high-quality preference datasets to generalize effectively.

On the other hand, inference-phase alignment methods apply constraints during decoding to regulate model behavior without modifying its parameters such as controlled decoding techniques, prompt-based filtering, and output rejection sampling (Guo et al., 2023; Li et al., 2023b; Zou et al., 2024) to obtain desired output. They suffer from several limitations such as increased computational overhead during inference, susceptibility to adversarial attacks and less reliable for long-term alignment compared to fine-tuning-based approaches.

2.2 Token-level Preference Learning

Preference learning is conventionally applied at the sequence level, where alignment strategies optimize entire model outputs based on high-level human preferences. However, several studies have demonstrated that preference adherence is highly dependent on token-level interactions, which influence factors such as fluency, coherence, factual consistency, and adversarial robustness.

To address this, several approaches have incorporated token-level loss functions into fine-tuning-based preference learning. These approaches aim to improve alignment by optimizing token distributions rather than full-sequence outputs. PPO-max (Zheng et al., 2023) introduces a token-level KL penalty to control model divergence from preferred outputs. Deep alignment (Qi et al., 2024) shows that LLMs exhibit sensitivity to the placement of harmful prefixes and suffixes, which can significantly alter their response behavior. DEFT (Zhu et al., 2024) adjusts the model's output distribution through weighted token selection which is formulated as a dictionary-based preference representation by computing token frequency differences across preferred and non-preferred outputs.

Despite these advancements, existing token-level preference learning approaches still face several limitations: (1) They focus on local token probabilities but fail to model global semantic relationships across sequences. (2) Many methods depend on handcrafted heuristic functions, introducing biases that require extensive hyperparameter tuning. (3) Token-level constraints often lack distributional perspective, limiting their effectiveness in generalizing to structured preferences over longer contexts.

These challenges highlight the need for a principled, distribution-aware approach that integrates token-level alignment with global preference modeling. Optimal Transport (OT) offers a promising direction by enabling fine-tuning-based preference learning that aligns full-sequence distributions while preserving semantic consistency.

3 Methodology

In this section, we formulate the fine-tuning-based preference learning problem as an Optimal Transport problem. Preliminaries about Optimal Transport can be found in Appendix A .

3.1 Problem Definition

In general, the preference dataset is as follows:

$$D = \left\{ \left(x^{(i)}, y_+^{(i)}, y_-^{(i)} \right) \right\}_{i=1}^{N} \tag{1}$$

where $x^{(i)}$ represents the user query, $y_+^{(i)}$ represents the preferred answer, $y_-^{(i)}$ represents the non-preferred answer, and N is the total number of samples. Such data can be used for reward modeling or directly fine-tuned via various methods.

We assume that the model output distribution during the fine-tuning process is $\mathcal{Q}_\theta$, and there exists a distribution $\mathcal{P}$ that represents the target

preference information. In order for the model to conduct preference learning from the perspective of distribution, we aim to preserve the original form of the model's distribution while considering the semantic relationships between tokens to achieve global optimization. To do this, we quantify the gap between $\mathcal{Q}_\theta$ and $\mathcal{P}$, which is defined as the optimal transport problem from $\mathcal{Q}_\theta$ to $\mathcal{P}$, as shown in Equation 15. The preference difference is the minimum transport distance between them, denoted as $\mathcal{L}_{\text{PLOT}}$, which is incorporated into the fine-tuning methods' loss function $\mathcal{L}_{\text{vanilla}}$ as follows:

$$\mathcal{L} = \mathcal{L}_{\text{vanilla}} + \alpha \mathcal{L}_{\text{PLOT}} \tag{2}$$

where α is a hyperparameter that controls the weight of $\mathcal{L}_{\text{PLOT}}$ in the overall loss. Before this, we first need to derive the target preference distribution $\mathcal{P}$ and the elements c_{ij} used to construct the cost matrix C.

3.2 Preference Distribution

We denote the distribution containing preference information as the target preference distribution $\mathcal{P}_\text{t}$. This is the object that the model's output distribution is transported to, and it serves as the target for the model to learn the preference gap between them. It can be the output distribution $\mathcal{Q}_{\text{rm}}$ of a reward model, or, as in previous work (Zhu et al., 2024), a dictionary $\mathcal{Q}_{\text{diff}}$ consisting of the difference between the token frequencies of positive and negative examples. In essence, we require a distribution that embodies preference information and apply the following operations $\Phi(\cdot)$:

$$\Phi(\mathcal{P}_\text{t}) = \frac{T(p_i)}{\sum_{j=1}^{n} T(p_j)}, \quad T : \mathbb{R} \to \mathbb{R}_+ \tag{3}$$

where p_i represents the value of $\mathcal{P}_\text{t}$ at token$_i$, T is an arbitrary non-negative function, and n denotes the dimension of $\mathcal{P}_\text{t}$, typically the size of the vocabulary. The purpose of this step is to transform $\mathcal{P}_\text{t}$ into a strict mathematical distribution, enabling its participation in subsequent OT calculations.

3.3 Token Embedding

Once the two distributions, $\mathcal{Q}_\theta$ and $\mathcal{P}_\text{t}$, for the OT problem are obtained, the default cost matrix C can be used to solve the problem, where the cost of tokens in the same position is 0, and the cost of tokens in different positions is 1. This approach computes the minimal cost, where the distance between tokens is not considered, and the cost is calculated solely based on the token values in $\mathcal{Q}_\theta$ and $\mathcal{P}_\text{t}$. However, in preference learning tasks, tokens carry rich semantic information. Since OT calculations provide the cost matrix C to incorporate such information, we extract the model's embedding table $\mathbf{E}$ of all tokens in the semantic space:

$$\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n] \tag{4}$$

where each sub-vector $\mathbf{e}_i$ represents the embedding of the i-th token, and n denotes the size of the vocabulary. To simplify the computational complexity and unify the dimensions, we apply an l-norm mapping to each sub-vector $\mathbf{e}_i$ in the embedding space, bringing them into a specific distance space:

$$c_{ij} = |\|\mathbf{e}_i\|_l - \|\mathbf{e}_j\|_l| \tag{5}$$

in which the distance metric l can be arbitrarily chosen. This results in a cost matrix C that incorporates rich semantic information, enabling us to account for the semantic relationships between tokens. This ultimately leads to improved preference learning.

3.4 Minimal Distance

Given $\mathcal{Q}$, $\mathcal{P}$, and C, we can proceed with solving the OT problem from Equation 15: $\mathcal{Q}$ represents the model output distribution $\mathcal{Q}_\theta$; the selection of $\mathcal{P}$ is considered based on previous work (Zhu et al., 2024) that constructs $\mathcal{Q}_{\text{diff}}$ from preference data, which effectively extracts preference information. We use it as a candidate for $\mathcal{P}_\text{t}$:

$$\mathcal{Q}_{\text{diff}} = \frac{\mathcal{Q}_+}{\sum \mathcal{Q}_+} - \frac{\mathcal{Q}_-}{\sum \mathcal{Q}_-} \tag{6}$$

where $Q_{+/-}$ is the token frequency of all $y_{+/-}$, respectively. However, considering that the difference between the two distributions lies in the range $[-1, 1]$ and is not a strict mathematical distribution, to preserve the token-wise differences in values and maintain the form of $\mathcal{Q}_{\text{diff}}$ itself, we apply a non-negative transformation by subtracting the minimum value as T:

$$T(Q_{\text{diff}}) = Q_{\text{diff}} - \min(Q_{\text{diff}}) \tag{7}$$

then the range of values for $\mathcal{Q}_{\text{diff}}$ becomes $[0, \max(Q_{\text{diff}}) - \min(Q_{\text{diff}})]$. After normalization via Equation 3, a strict target preference distribution $\mathcal{P}_\text{t}$ is obtained. For the cost matrix, we set $l = 2$, which corresponds to the $\mathbf{L}_2$ norm, to obtain the Euclidean distance of each token from the origin:

$$\|\mathbf{e}\|_2 = \sqrt{\sum_{i=1}^{d} e_i^2} \tag{8}$$

where d is the dimension of embedding vectors. We then transform $\mathbf{E}$ into a one-dimensional vector with the same length as $\mathcal{Q}_\theta$ and $\mathcal{P}_\mathrm{t}$, and compute all the elements of the cost matrix using Equation 5. Thus, Equation 15 can be written as the new loss item $\mathcal{L}_{\text{PLOT}}$:

$$\begin{aligned}
&\mathcal{L}_{\text{PLOT}}(\mathcal{Q}_\theta, \mathcal{P}_\mathrm{t}) \\
&= \min_{\Gamma \in \Pi(\mathcal{Q}_\theta, \mathcal{P}_\mathrm{t})} \sum_{i,j} |\|\mathbf{e}_i\|_2 - \|\mathbf{e}_j\|_2| \gamma_{ij} \\
&= \min_{\Gamma \in \Pi(\mathcal{Q}_\theta, \mathcal{P}_\mathrm{t})} \langle C, \Gamma \rangle \\
&\qquad \text{s.t. } \Gamma \mathbf{1} = \mathcal{Q}_\theta, \quad \Gamma^\top \mathbf{1} = \mathcal{P}_\mathrm{t}, \\
&\qquad\qquad \gamma_{ij} \geq 0 \quad \forall i, j.
\end{aligned} \tag{9}$$

By solving this constrained linear programming problem, we can obtain the minimum transport cost between $\mathcal{Q}_\theta$ and $\mathcal{P}_\mathrm{t}$ at each step by Γ. However, in practice, the vocabulary size of LLMs is typically large and the cost matrix and constraints make the problem difficult to solve. Based on the previous derivation, we have obtained one-dimensional discrete vectors $\mathcal{Q}_\theta$, $\mathcal{P}_\mathrm{t}$, and $\mathbf{E}$ of equal length. Therefore, the solution to Equation 9 is equivalent to the computation of the one-dimensional Wasserstein distance, defined as $W_1(\mathcal{Q}, \mathcal{P})$ (Villani et al., 2009; Peyré et al., 2019):

$$\begin{aligned}
W_1(\mathcal{Q}, \mathcal{P}) &= \int_{-\infty}^{\infty} |F_q(x) - F_p(x)| \, dx \\
&= \sum_{i}^{n-1} |F_q(x_i) - F_p(x_i)| \Delta x_i
\end{aligned} \tag{10}$$

where $F_{\text{II}}(x)$ represents the Cumulative Distribution Function (CDF) of distribution $\mathcal{Q}$:

$$F_q(x_i) = \sum_{t \leq x_i} \mathcal{Q}(X = t) \tag{11}$$

Δx represents the difference between adjacent x values:

$$\Delta x_i = x_{i+1} - x_i \tag{12}$$

In our case, this corresponds to the distance between two tokens. Thus, the final calculation for $\mathcal{L}_{\text{PLOT}}$ becomes:

$$\begin{aligned}
&\mathcal{L}_{\text{PLOT}}(\mathcal{Q}_\theta, \mathcal{P}_\mathrm{t}) = W_1(\mathcal{Q}_\theta, \mathcal{P}_\mathrm{t}) \\
&= \sum_{i}^{n-1} |F_{q_\theta}(x_i) - F_{p_\mathrm{t}}(x_i)| \Delta x_i.
\end{aligned} \tag{13}$$

The minimal distance between $\mathcal{Q}_\theta$ and $\mathcal{P}_\mathrm{t}$ causes the model's overall output distribution to align more closely with the preference distribution, especially for those tokens that most align with or deviate from the preference. Additionally, by considering the embedding vectors, PLOT prioritizes transportation between tokens that are close in the semantic space, making the model consider not just the transport distance between individual tokens, but also all tokens in the semantic space, achieving a form of global optimization.

4 Experiments

This paper focuses on aligning the model with preferences for **Harmless** content and employs a series of jailbreak methods to assess the model's resistance to attacks. Additionally, we conducted evaluations to measure the model's general capabilities.

4.1 Training Details

Data We used the previously refined HH-RLHF dataset (Bai et al., 2022a) from the prior work PRO (Song et al., 2024) which includes higher-quality ChatGPT [1] responses added to all samples as our training data. We extracted the **Harmless$_{\text{base}}$** subset which contains 42,536 samples from it and constructed the preference distribution $\mathcal{P}_\mathrm{t}$ using the positive and negative examples from the enhanced dataset. Considering training costs, we randomly selected 4,000 of the data for training.

Method Considering training costs and the rigor of the method, we selected the classic DPO method to validate the effectiveness of $\mathcal{L}_{\text{PLOT}}$.

Model We use the latest Llama3.2-3B-Instruct [2] as our base model, denoted as **Instruct**. The model trained on the training set with DPO is referred to as **DPO**. Since this paper only conducts enhancement experiments using the DPO method, for clarity, we refer to the model trained with PLOT added on top of DPO as **PLOT**.

Setup All experiments were conducted on 4 NVIDIA A100 80GB GPUs, with a batch size of 1 per GPU and a total batch size of 4. The training was performed for 1 epoch with the hyperparameter $\beta = 0.1$ for DPO and $\alpha = 8$ for PLOT. The training duration of DPO was approximately 487.03 seconds, while for PLOT, it was approximately 500.80 seconds.

[1] https://chat.openai.com/

[2] https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct

Methods		Behaviors									Total		
		Standard			Copyright			Contextual					
		Instruct	DPO	PLOT	Instruct	DPO	PLOT	Instruct	DPO	PLOT	Instruct	DPO	PLOT
ZS	$n = 1$	26.33 ±3.06	4.17 ±0.85	1.00 ±0.41	13.00 ±1.41	6.00 ±0.82	5.00 ±0.82	50.00 ±4.55	18.00 ±1.41	8.00 ±0.82	28.92 ±2.54	8.08 ±0.85	3.75 ±0.41
	$n = 500$	27.28 ±1.41	3.50 ±0.43	0.47 ±0.05	12.97 ±0.21	9.20 ±0.37	6.33 ±0.66	57.22 ±1.88	17.63 ±2.09	10.30 ±1.56	31.18 ±1.12	8.46 ±0.81	4.39 ±0.20
PEZ	$n = 5$ $T = 500$	21.87 ±0.90	5.37 ±0.33	4.73 ±0.12	22.60 ±0.16	7.13 ±0.98	4.20 ±0.43	20.27 ±1.64	3.93 ±0.52	3.27 ±0.62	21.65 ±0.54	5.45 ±0.25	4.23 ±0.20
GBDA	$n = 5$ $T = 500$	22.33 ±0.66	5.03 ±0.54	4.87 ±0.21	20.27 ±1.09	6.33 ±1.91	4.53 ±0.34	16.60 ±1.23	7.00 ±1.45	3.00 ±0.71	20.38 ±0.53	5.85 ±0.37	4.32 ±0.13
UAT	$n = 1$ $k = 3$ $T = 100$	20.00 ±2.04	7.67 ±1.43	5.17 ±0.24	21.00 ±1.63	8.33 ±1.70	7.00 ±0.82	17.67 ±0.47	3.33 ±0.94	2.33 ±0.47	19.67 ±0.51	6.75 ±0.20	4.92 ±0.24
SFS	$k = 5$ $T = 50$	39.22 ±1.84	25.00 ±2.04	16.67 ±1.55	42.33 ±3.86	31.00 ±0.00	17.67 ±1.25	32.67 ±6.24	22.00 ±2.16	16.67 ±1.25	38.42 ±1.36	25.75 ±1.24	16.92 ±0.51
GCG	$n = 1$ $T = 500$	48.50 ±4.30	30.67 ±2.87	26.67 ±1.70	51.00 ±3.56	31.67 ±0.94	27.67 ±1.25	40.67 ±4.19	27.33 ±1.70	26.33 ±0.94	47.17 ±1.43	30.08 ±0.82	26.83 ±0.42

Table 1: The experimental results for the red reaming attack. Attack Success Rate (ASR) is used to assess the defense capability of different models under various attack methods. PLOT demonstrates significant performance enhancements over DPO across all attack methods, showing improvements in each test data subset as well as the overall performance with better stability.

4.2 Red Teaming Attack

We used HarmBench (Mazeika et al., 2024), which is a standardized evaluation framework for various automated red teaming attack methods, to supports a multidimensional evaluation of the model's defense capabilities through both functional and semantic classification. In terms of the testing data, it integrates current harmful content datasets and manually designs harmful behaviors to ensure that they violate legal or widely accepted norms. It includes seven categories of harmful content, and excluding multimodal data, the pure text data can be categorized into three types, as shown in Table 2. **Attack Success Rate (ASR)** was used as the evaluation metric, with a fine-tuned Llama2-13B-Chat [3] model serving as the classifier to determine whether an attack was successful.

Behaviors	#Samples	Source & Description
Standard	200	Based on AdvBench (2023) and TDC 2023 (2023)
Copyright	100	Manually crafted requests for copyrighted content
Contextual	100	Manually crafted complex requests with context
Total	400	Manually filtered to ensure clearly harmful with no legitimate use

Table 2: Distribution and description of the HarmBench evaluation dataset.

Methods We selected a subset of red-team attack methods to test the model's defense capabilities against harmful content under various conditions. **Zero-Shot** (Perez et al., 2022) or **ZS** directly generates n cases for each behavior, resulting in a total of $400 \times n$ test samples. For each behavior, **PEZ** (Wen et al., 2024) generates n cases with an optimized embedding vector to induce harmful content, totaling $400 \times n$, iterated for T rounds, using only the target model. **GBDA** (Guo et al., 2021) uses the Gumbel-Softmax technique for the target model to convert discrete token selection into a differentiable operation, with n cases generated for each behavior, iterating for T steps. **UAT** (Wallace et al., 2019) generates adversarial trigger tokens via gradient-based optimization for each behavior. Each case is iterated for T rounds, selecting tokens from the top k candidates, n cases are generated, totaling $400 \times n$. **Sophistic Few-Shot** (Perez et al., 2022) or **SFS** generates candidate prompts per iteration using updated k-shot examples, refining over T iterations, with the best candidate selected as the final prompt. For each behavior, **GCG** (Zou et al., 2023) optimizes an adversarial suffix at the token level, with n cases generated, iterating for T

[3] https://huggingface.co/cais/HarmBench-Llama-2-13b-cls

rounds, resulting in a total of $400 \times n$ cases, and only the target model needs to be loaded.

Main Result To fully validate the effectiveness of PLOT, for each red team attack method, we conducted 3 times of experiment on each fine-tuned model and under each hyperparameter setting, averaged the results, and calculated the standard deviation. The default Mixtral-8x7B-Instruct-v0.1 [4] from HarmBench was chosen as the attack model for some of the methods.

As shown in Table 1, we compared the ASR of the Instruct model, the DPO fine-tuned model, and the PLOT enhanced model under each method. Under each experimental setting and for every method, PLOT achieved varying degrees of reduction in ASR compared to DPO: it reduced ASR by 1.22-1.83% for the PEZ, GBDA, and UAT methods; by 3.25-4.33% for the Zero-shot and GCG methods; and by a significant 8.83% for the SFS method. Except for the slightly higher standard deviation (0.24) for the UAT method compared to DPO (0.20), the standard deviations for other methods are lower than those of Instruct and DPO. This demonstrates that the model under PLOT optimization exhibits significantly enhanced defense capabilities and stability against various attack methods.

In addition, we also plotted the line charts of ASR variations for different values of n in Zero-Shot and for different update steps T in GCG, as shown in Figure 1. It can be seen that PLOT further enhances the defense capability over DPO while exhibiting stronger stability against attacks, consistent with the conclusions from Table 1.

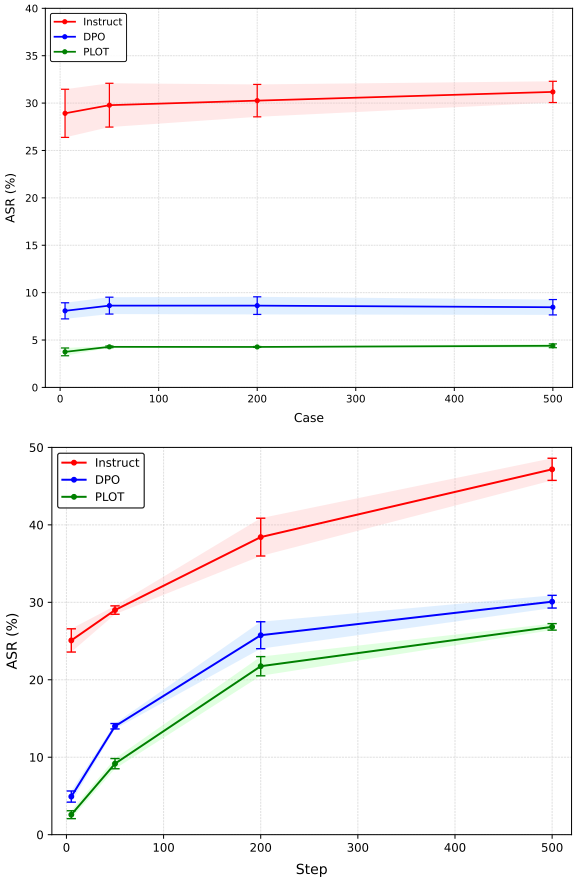

Figure 1: The ASR curves of three models under different case counts for the Zero-Shot method (Up) and varying update steps of GCG (Down). PLOT consistently demonstrates superior defense capabilities and stability compared to DPO.

4.3 General Capabilities

While the model demonstrates improved preference learning, it is crucial to assess the impact on the model's general capabilities. Therefore, we employed AlpacaEval (Li et al., 2023a) to comprehensively evaluate the effect of $\mathcal{L}_{\text{PLOT}}$ on the model's general output quality. We followed the AlpacaEval 2.0 setup, with GPT-4 as the reference for comparing response quality and scoring, and utilized Length-controlled (LC) Win Rate (Dubois et al., 2024) as the evaluation metric, as shown in Figure 2.

As shown, after DPO fine-tuning, the overall response quality of the model declined to some extent, with the LC Win Rate dropping from $17.93\% \pm 1.24\%$ to $13.64\% \pm 1.11\%$, which may be due to the random sampling of the training data. However, after incorporating PLOT with DPO, the model's response quality did not worsen, increasing slightly $14.06\% \pm 1.09\%$, indicating that PLOT did not exacerbate the negative impact of the original fine-tuning method on response quality. In other words, the distribution-based loss term did not interfere with the model's normal output distribution, thereby preserving the quality of the model's responses under fine-tuning, which corresponds to its general capabilities.

4.4 Ablation and Analysis

In this section, we conduct jailbreak experiments under the same setup as Section 4 and experiments are performed three times, with the average values and standard deviations computed. We compare the ASR results of different methods and settings to validate and analyze the effectiveness of $\mathcal{L}_{\text{PLOT}}$.

[4] https://huggingface.co/mistralai/Mixtral-8x7B-Instruct-v0.1

Methods	ZS	PEZ	GBDA	UAT	SFS	GCG
	n=500	n=5, T=500	n=5, T=500	n=1, k=3, T=100	k=5, T=50	n=1, T=500
DPO	8.46 ± 0.81	5.45 ± 0.25	5.85 ± 0.37	6.75 ± 0.20	25.75 ± 1.24	30.08 ± 0.82
+$\mathcal{R}_\mathcal{Q}$ (DEFT)	5.65 ± 0.61	5.08 ± 0.51	5.27 ± 0.41	5.25 ± 0.20	19.25 ± 1.27	28.42 ± 1.12
+$\mathcal{L}_{\text{PLOT}}$ w/o $\mathbf{E}$	4.91 ± 0.30	4.83 ± 0.17	4.40 ± 0.12	5.17 ± 0.12	17.83 ± 0.31	27.25 ± 0.54
+$\mathcal{L}_{\text{PLOT}}$	4.39 ± 0.20	4.23 ± 0.20	4.32 ± 0.13	4.92 ± 0.24	16.92 ± 0.51	26.83 ± 0.42

Table 3: A comparison of ASR across different loss components and experimental settings under various attack methods reveals that transitioning from DEFT to OT-formulated problem yields performance improvements, which are further enhanced by the inclusion of token embeddings.

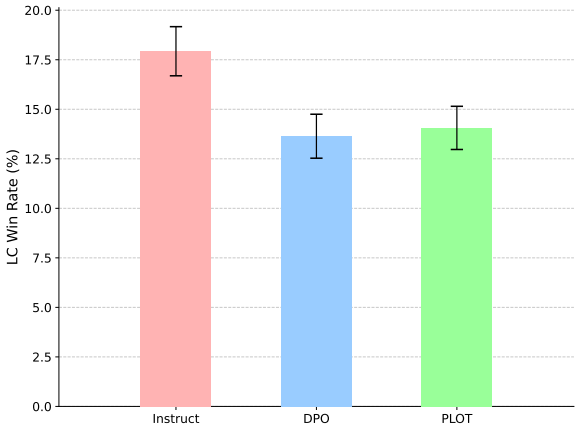

Figure 2: Comparison of the Length-controlled (LC) Win Rate across the three models shows that $\mathcal{L}_{\text{PLOT}}$ preserves the general capabilities of the model under the original fine-tuning method.

Effectiveness of OT Formulation Previous work DEFT (Zhu et al., 2024) introduced the distribution reward $\mathcal{R}_\mathcal{Q}$, where the term $\mathcal{Q}_{\text{diff}}$, derived from Equation 6, is element-wise multiplied by the model output $\mathcal{Q}_\theta$ at the token level and then summed as follows:

$$\mathcal{R}_\mathcal{Q} = \sum \mathcal{Q}_{\text{diff}} \odot \log \mathcal{Q}_\theta \tag{14}$$

and this reward is subsequently incorporated as a new loss term in the fine-tuning procedure. As previously noted, the value range for each token position lies within $[-1, 1]$ and it does not satisfy the conditions of a true mathematical distribution. As illustrated in Table 3, although the inclusion of $\mathcal{R}_\mathcal{Q}$ leads to promising results, particularly in the context of $\mathcal{Q}_{\text{diff}}$ effectively extracting preference information, its performance in defending against various red team attack methods remains inferior compared to the approach we propose. This highlights the effectiveness of reformulating the preference learning problem at the distribution level as an optimal transport (OT) problem for its resolution. Since the operation in Equation 14 is an empirical approach, it essentially focuses on local optimization of individual tokens, whereas the solution to the OT problem offers a global optimization from the perspective of the entire distribution.

Efficacy of Embedding In order to investigate the practical effect of extracting token embeddings for computing the cost matrix C, we discarded the embedding vector $\mathbf{E}$ and replaced the cost matrix with the default 0-1 cost matrix as described in Section 3.3. The new loss term is denoted as $\mathcal{L}_{\text{PLOT}}$ w/o $\mathbf{E}$, and DPO training was conducted under the same setup. As shown in Table 3, the model excluding token embeddings consistently achieved higher ASR across multiple red teaming attack methods compared to the standard $\mathcal{L}_{\text{PLOT}}$. This clearly demonstrates that incorporating token embeddings for inter-token distance computation inherently leverages semantic space relationships, which, in theory, enables more sophisticated distribution-level optimization through richer information. These performance improvements are in line with the theoretical derivations.

5 Conclusion

In this paper, we introduce PLOT, a novel loss term designed to enhance the effectiveness of preference learning in fine-tuning alignment methods. By formulating token-level preference learning as an Optimal Transport (OT) problem at the distribution level, PLOT effectively captures both the preference discrepancies between distributions and the rich semantic information encoded in the tokens. Experimental results on preferences for harmless content demonstrate that PLOT significantly improves the preference learning performance of the baseline method, while preserving the overall response quality of the model. This work contributes a new approach to enhancing preference learning in fine-tuning methods, offering both theoretical insights and practical improvements in the field.

Limitations

First, we introduce a preference learning enhancement loss based on fine-tuning methods. However, considering the challenges in testing and quantifying preferences, we selected the **Harmful/Harmless** preference, which is both fundamental and widely adopted in current evaluation practices, to demonstrate the effectiveness of our approach. Future work will involve testing $\mathcal{L}_{\text{PLOT}}$ on other types of preferences, such as logical reasoning in mathematical sciences and correctness and complexity in code-related tasks, to further establish its generalizability. Second, due to time and cost constraints, our baseline fine-tuning method only considers **DPO**, with the base model being **Llama-3.2-3B-Instruct**. The training data consists of 4,000 randomly selected samples from the $\text{Harmless}_{\text{base}}$ subset of the HH-RLHF dataset. Further experiments with larger and more diverse datasets, methods, and models are necessary for enhanced comparative analysis. Finally, several custom components in our method, such as the selection of the preference distribution $\mathcal{P}_{\text{t}}$, the choice of the non-negativity function T, and the selection of the embedding norm l, offer potential points for further experimental investigation.

References

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

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

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

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022b. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Sebastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, et al. 2023. Alpagasus: Training a better alpaca with fewer data. *arXiv preprint arXiv:2307.08701*.

Brian Christian. 2021. *The alignment problem: How can machines learn human values?* Atlantic Books.

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

Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26.

Gelei Deng, Yi Liu, Yuekang Li, Kailong Wang, Ying Zhang, Zefeng Li, Haoyu Wang, Tianwei Zhang, and Yang Liu. 2023. Jailbreaker: Automated jailbreak across multiple large language model chatbots. *arXiv preprint arXiv:2307.08715*.

Yi Dong, Zhilin Wang, Makesh Narsimhan Sreedhar, Xianchao Wu, and Oleksii Kuchaiev. 2023. Steerlm: Attribute conditioned sft as an (user-steerable) alternative to rlhf. *arXiv preprint arXiv:2310.05344*.

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

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

Iason Gabriel. 2020. Artificial intelligence, values, and alignment. *Minds and machines*, 30(3):411–437.

Iason Gabriel and Vafa Ghazavi. 2021. The challenge of value alignment: From fairer algorithms to ai safety. *arXiv preprint arXiv:2101.06060*.

Yuxuan Gu, Wenjie Wang, Xiaocheng Feng, Weihong Zhong, Kun Zhu, Lei Huang, Tat-Seng Chua, and Bing Qin. 2024. Length controlled generation for black-box llms. *arXiv preprint arXiv:2412.14656*.

Chuan Guo, Alexandre Sablayrolles, Hervé Jégou, and Douwe Kiela. 2021. Gradient-based adversarial attacks against text transformers. *arXiv preprint arXiv:2104.13733*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Geyang Guo, Ranchi Zhao, Tianyi Tang, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Beyond imitation: Leveraging fine-grained quality signals for alignment. *arXiv preprint arXiv:2311.04072*.

Jian Hu, Li Tao, June Yang, and Chandler Zhou. 2023. Aligning language models with offline reinforcement learning from human feedback. *arXiv preprint arXiv:2308.12050*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Zachary Kenton, Tom Everitt, Laura Weidinger, Iason Gabriel, Vladimir Mikulik, and Geoffrey Irving. 2021. Alignment of language agents. *arXiv preprint arXiv:2103.14659*.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Lu, Thomas Mesnard, Colton Bishop, Victor Carbune, and Abhinav Rastogi. 2023. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*.

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

Yuhui Li, Fangyun Wei, Jinjing Zhao, Chao Zhang, and Hongyang Zhang. 2023b. Rain: Your language models can align themselves without finetuning. *arXiv preprint arXiv:2309.07124*.

Ziniu Li, Tian Xu, Yushun Zhang, Yang Yu, Ruoyu Sun, and Zhi-Quan Luo. 2023c. Remax: A simple, effective, and efficient method for aligning large language models. *arXiv preprint arXiv:2310.10505*.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024a. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Tianqi Liu, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter J Liu, and Jialu Liu. 2023. Statistical rejection sampling improves preference optimization. *arXiv preprint arXiv:2309.06657*.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2024b. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. In *The Twelfth International Conference on Learning Representations*.

Mantas Mazeika, Dan Hendrycks, Huichen Li, Xiaojun Xu, Sidney Hough, Andy Zou, Arezoo Rajabi, Qi Yao, Zihao Wang, Jian Tian, et al. 2023. The trojan detection challenge. In *NeurIPS 2022 Competition Track*, pages 279–291. PMLR.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, et al. 2024. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. *arXiv preprint arXiv:2402.04249*.

Tetsuro Morimura, Mitsuki Sakamoto, Yuu Jinnai, Kenshi Abe, and Kaito Air. 2024. Filtered direct preference optimization. *arXiv preprint arXiv:2404.13846*.

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.

Arka Pal, Deep Karkhanis, Samuel Dooley, Manley Roberts, Siddartha Naidu, and Colin White. 2024. Smaug: Fixing failure modes of preference optimisation with dpo-positive. *arXiv preprint arXiv:2402.13228*.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*.

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. 2022. Red teaming language models with language models. *arXiv preprint arXiv:2202.03286*.

Gabrijela Perković, Antun Drobnjak, and Ivica Botički. 2024. Hallucinations in llms: Understanding and addressing challenges. In *2024 47th MIPRO ICT and Electronics Convention (MIPRO)*, pages 2084–2088. IEEE.

Gabriel Peyré, Marco Cuturi, et al. 2019. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607.

Xiangyu Qi, Ashwinee Panda, Kaifeng Lyu, Xiao Ma, Subhrajit Roy, Ahmad Beirami, Prateek Mittal, and Peter Henderson. 2024. Safety alignment should be made more than just a few tokens deep. *arXiv preprint arXiv:2406.05946*.

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

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

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

Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. 2024. " do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 1671–1685.

Prasann Singhal, Nathan Lambert, Scott Niekum, Tanya Goyal, and Greg Durrett. 2024. D2po: Discriminator-guided dpo with response evaluation models. *arXiv preprint arXiv:2405.01511*.

Feifan Song, Bowen Yu, Minghao Li, Haiyang Yu, Fei Huang, Yongbin Li, and Houfeng Wang. 2024. Preference ranking optimization for human alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18990–18998.

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

Cédric Villani et al. 2009. *Optimal transport: old and new*, volume 338. Springer.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing nlp. *arXiv preprint arXiv:1908.07125*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Yuxin Wen, Neel Jain, John Kirchenbauer, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2024. Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery. *Advances in Neural Information Processing Systems*, 36.

Frank F Xu, Uri Alon, Graham Neubig, and Vincent Josua Hellendoorn. 2022. A systematic evaluation of large language models of code. In *Proceedings of the 6th ACM SIGPLAN International Symposium on Machine Programming*, pages 1–10.

Jian Yang, Jiaxi Yang, Ke Jin, Yibo Miao, Lei Zhang, Liqun Yang, Zeyu Cui, Yichang Zhang, Binyuan Hui, and Junyang Lin. 2024. Evaluating and aligning codellms on human preference. *arXiv preprint arXiv:2412.05210*.

Sibo Yi, Yule Liu, Zhen Sun, Tianshuo Cong, Xinlei He, Jiaxing Song, Ke Xu, and Qi Li. 2024. Jailbreak attacks and defenses against large language models: A survey. *arXiv preprint arXiv:2407.04295*.

Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. 2023. Rrhf: Rank responses to align language models with human feedback. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Rui Zheng, Shihan Dou, Songyang Gao, Yuan Hua, Wei Shen, Binghai Wang, Yan Liu, Senjie Jin, Qin Liu, Yuhao Zhou, et al. 2023. Secrets of rlhf in large language models part i: Ppo. *arXiv preprint arXiv:2307.04964*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2023. Lima: Less is more for alignment. *arXiv preprint arXiv:2305.11206*.

Liang Zhu, Feiteng Fang, Yuelin Bai, Longze Chen, Zhexiang Zhang, Minghuan Tan, and Min Yang. 2024. Deft: Distribution-guided efficient fine-tuning for human alignment. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15318–15331.

Terry Yue Zhuo, Minh Chien Vu, Jenny Chim, Han Hu, Wenhao Yu, Ratnadira Widyasari, Imam Nur Bani Yusuf, Haolan Zhan, Junda He, Indraneil Paul, et al. 2024. Bigcodebench: Benchmarking code generation with diverse function calls and complex instructions. *arXiv preprint arXiv:2406.15877*.

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

Andy Zou, Long Phan, Justin Wang, Derek Duenas, Maxwell Lin, Maksym Andriushchenko, J Zico Kolter, Matt Fredrikson, and Dan Hendrycks. 2024. Improving alignment and robustness with circuit breakers. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

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

A Preliminary

A.1 Optimal Transport

Optimal Transport (OT) provides a principled way to compare probability distributions by computing the minimum cost required to transform one distribution into another. Unlike traditional divergence measures such as Kullback-Leibler (KL) divergence, which compares probability distributions in terms of relative entropy, OT explicitly models the movement of probability mass, making it particularly effective for structured alignment problems.

Given two probability distributions, $\mathcal{Q}$ and $\mathcal{P}$, the **Optimal Transport (OT) Problem** is defined as follows (Peyré et al., 2019):

$$\begin{aligned}\mathbf{OT}(\mathcal{Q},\mathcal{P}) &= \min_{\Gamma\in\Pi(\mathcal{Q},\mathcal{P})} \langle C,\Gamma\rangle \\ &= \min_{\Gamma\in\Pi(\mathcal{Q},\mathcal{P})} \sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij}\gamma_{ij} \\ \text{s.t. } \Pi(\mathcal{Q},\mathcal{P}) &= \left\{ \Gamma\in\mathbb{R}_{+}^{n\times m} \;\middle|\; \begin{matrix}\Gamma\mathbf{1}_m = \mathcal{Q},\\ \Gamma^{\top}\mathbf{1}_n = \mathcal{P}\end{matrix} \right\}.\end{aligned} \tag{15}$$

Here, C represents the cost matrix, where each element c_{ij} quantifies the transport cost between point q_i in $\mathcal{Q}$ and point p_j in $\mathcal{P}$. The transport plan Γ is a joint probability matrix, where each element γ_{ij} represents the amount of mass transported from q_i to p_j. The constraints enforce that:

- The total transported mass from each point in $\mathcal{Q}$ must equal its original mass.
- The total mass arriving at each point in $\mathcal{P}$ must match its target distribution.
- Each element γ_{ij} in Γ must be non-negative.

The objective is to determine an optimal transport plan Γ that minimizes the overall transport cost.

A.2 Relevance of Optimal Transport to Preference Learning

Optimal Transport has been widely used in distributional alignment tasks, including generative modeling and domain adaptation (Arjovsky et al., 2017). In the context of preference learning, OT provides a natural way to measure the discrepancy between a model's predicted distribution and an ideal preference distribution. Unlike token-level objectives, which optimize local probability assignments independently, OT considers global structure within the output distribution, ensuring more stable and context-aware preference learning.

While OT is a powerful framework, solving it exactly is computationally expensive, typically requiring $O(n^3 \log n)$ operations. To address this, regularized OT variants such as entropy-regularized OT (Cuturi, 2013) have been proposed, which introduce an entropy penalty to encourage smoother transport plans and reduce computational complexity to $O(n^2)$. These approximations make OT feasible for large-scale preference learning applications.

By leveraging OT in fine-tuning-based preference learning, models can learn to align with human preferences more robustly while preserving semantic coherence across generated outputs.