# Exposing Privacy Gaps: Membership Inference Attack on Preference Data for LLM Alignment

**Qizhang Feng*** **Siva Rajesh Kasa*** **Santhosh Kasa**

**Hyokun Yun** **Choon Hui Teo** **Sravan Bodapati**

Amazon Inc.

## Abstract

Large Language Models (LLMs) have seen widespread adoption due to their remarkable natural language capabilities. However, when deploying them in real-world settings, it is important to align LLMs to generate texts according to acceptable human standards. Methods such as Proximal Policy Optimization (PPO) and Direct Preference Optimization (DPO) have enabled significant progress in refining LLMs using human preference data. However, the privacy concerns inherent in utilizing such preference data have yet to be adequately studied. In this paper, we investigate the vulnerability of LLMs aligned using two widely used methods - DPO and PPO - to membership inference attacks (MIAs). Our study has two main contributions: first, we theoretically motivate that DPO models are more vulnerable to MIA compared to PPO models; second, we introduce a novel reference-based attack framework specifically for analyzing preference data called PREMIA (Preference data MIA). Using PREMIA and existing baselines we empirically show that DPO models have a relatively heightened vulnerability towards MIA.

## 1 INTRODUCTION

Large language models (LLMs) have seen a surge in their adoption in the recent past due to their remarkable capabilities on a wide range of natural language processing (NLP) tasks such as question answering, code generation, etc (Zhao et al., 2023). When deployed in real-world scenarios, it is important to align LLMs to human preferences. Techniques such as Proximal Policy Optimization (PPO) and Direct Preference Optimization (DPO) play a key role in aligning LLMs with human ethical standards by leveraging human-derived preference data (Christiano et al., 2017; Rafailov et al., 2024; Yang et al., 2024). Although these approaches improve the alignment of models with human values, they are fraught with privacy concerns because of their use of human-generated data. In this work, we investigate the Membership Inference Attack (MIA), a widely-studied vulnerability that attempts to determine whether if specific data points are part of the model's preference dataset. The study of MIA highlights vulnerabilities in a variety of machine learning paradigms, including several recent studies that specifically focus on LLMs (Fu et al., 2023; Shi et al., 2024). Although existing research on MIA in the context of LLMs highlights the need to evaluate and address the need for privacy concerns, the unique challenges posed by alignment methods such as the PPO and DPO approaches (where preference data directly influences model behavior) remain to be explored. Traditional MIA frameworks fall short when applied to the complex, context-dependent optimization procedures used in LLM alignment. In this paper, we theoretically and empirically show that reward-model-free approaches such as DPO are more susceptible to MIA, as compared to the reward-model-based approaches such as PPO. We also introduce a novel reference-model based MIA framework that is specifically tailored towards preference data and LLM alignment, providing a more precise analysis tool that can better highlight these vulnerabilities. Our contributions to this field are twofold:

- **Comparative Vulnerability Assessment of DPO and PPO Models:** First, we theoretically motivate that despite being asymptotically equivalent in the optimization objective, DPO tends to overfit on the preference data vis à vis PPO. Next we derive a generic lower bound on the Bayes optimal membership on algorithms that tend to overfit, which shows why DPO models are

*These authors contributed equally to this work.
Contact: {qzf,kasasiva}@amazon.com

Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258. 

more vulnerable to MIA compared to PPO models..

- **Introduction of a Novel Reference-based Attack Framework:** Following up on the theoretical overfitting results, we empirically demonstrate the differential susceptibility of DPO vs PPO to MIA on two real world datasets. We propose PREMIA, a comprehensive optimistic attack framework tailored to assess the vulnerability of preference data to MIA, providing a practical upper bound to quantify the extent of attack effectiveness. Through PREMIA, along with existing MIA frameworks, our experiments highlight the complex interplay between model sizes and task difficulty in LLM alignment.

# 2 PRELIMINARIES

This section introduces the notations and background concepts required for the rest of the paper. We begin by defining the frameworks of PPO and DPO, followed by an overview of MIAs.

## 2.1 Model Alignment

Model alignment ensures LLMs adhere to human values and ethics by adjusting their outputs to match human preferences (Hendrycks et al., 2021; Ouyang et al., 2022). Such alignment is critical for creating AI systems that act in ways that benefit humans and reduce the risks associated with improper alignment. Among the various model alignment techniques, PPO and DPO are some of the widely used approaches Xu et al. (2024).

### 2.1.1 Proximal Policy Optimization (PPO)

Stiennon et al. (2020) and Bai et al. (2022) illustrate Reinforcement Learning from Human Feedback (RLHF) that integrates human feedback into the alignment of pre-trained Language Models (LMs), encompassing three phases: Supervised Fine-Tuning (SFT), Preference Sampling with Reward Learning, and Reinforcement Learning (RL) through PPO.

**SFT** begins the process by fine-tuning a pre-trained LM on task-specific data to obtain a model $\pi^{\mathrm{SFT}}$, enhancing the LLM's performance on the task at hand.

**Preference Data Collection** involves gathering a set of preference data pairs $(x, y_w, y_l)$, where $x$ is a prompt and $y_w$, $y_l$ are two different responses. Here, $y_w$ is the response preferred by human evaluators over $y_l$ for the given context $x$.

**Reward Modeling Phase** uses the preference pairs to train the reward model $r_\phi(x, y)$, where $\phi$ represents the trainable parameters. The trainable model can be a classification header layer attached to the base model or a separate model. The Bradley-Terry (BT) model is commonly used to represent the probability that one response is better than another:

$$\max_{\phi} -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log \sigma(r_\phi(x,y_w) - r_\phi(x,y_l))\right], \quad (1)$$

where $r_\phi(x, y)$ models the likelihood of preferring $y_w$ to $y_l$ given the prompt $x$, and $\mathcal{D}$ denotes the dataset of preference pairs. This loss function measures the accuracy of the reward model in predicting human preferences.

**RL Fine-Tuning Phase** then fine-tunes the LM further using the learned reward function, striving to align model outputs with human preferences while maintaining generative diversity:

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(y|x)}[r_\phi(x,y)] - \beta\mathbb{D}_{\mathrm{KL}}[\pi_\theta(y|x)||\pi^{\mathrm{SFT}}(y|x)], \quad (2)$$

balancing fidelity to human feedback with the preservation of the model's original capabilities. Here, $\pi_\theta$ represents the policy of the language model parameterized by $\theta$, the trainable parameters. The optimization in Equation 2 is carried out using Proximal Policy Optimization (PPO) method (Schulman et al., 2017).

### 2.1.2 Direct Preference Optimization (DPO)

DPO offers a simplified approach towards aligning language models by directly leveraging preference data, bypassing the explicit reward model construction typically associated with PPO methodologies (Rafailov et al., 2024). This method reformulates the two-step optimization procedure in Equations 1 and 2 into a single optimization problem that simultaneously optimizes the policy and encodes an implicit reward mechanism based on the preference data as follows:

Rafailov et al. (2024) show that the optimal policy that maximizes Equation 2 can be written as

$$\pi^*(y \mid x) = \frac{1}{Z(x)}\pi_{\mathrm{ref}}(y \mid x)\exp\left(\frac{1}{\beta}r_\phi(x,y)\right) \quad (3)$$

where $Z(x)$ is a partition function which depends on the prompt $x$ Rafailov et al. (2024); Xu et al. (2024). Rearranging Equation 3 gives the corresponding reward as follows:

$$r_\phi(x,y) = \beta\log\frac{\pi_\theta(y_w \mid x)}{\pi_{\mathrm{ref}}(y_w \mid x)} + C(x), \quad (4)$$

where $C : \mathcal{X} \to \mathbb{R}$ is a scalar function. Here, $\pi_{\mathrm{ref}}$ refers to a reference model which is typically chosen as the SFT model $\pi^{\mathrm{SFT}}$.

$$\max_{\pi_\theta} -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \quad (5)$$
$$\left[\log\sigma\left(\beta\log\frac{\pi_\theta(y_w \mid x)}{\pi_{\mathrm{ref}}(y_w \mid x)} - \beta\log\frac{\pi_\theta(y_l \mid x)}{\pi_{\mathrm{ref}}(y_l \mid x)}\right)\right].$$

This optimization method is often preferred over PPO because it simplifies training by optimizing directly on the preference data, which improves computational efficiency and is easier to implement (Rafailov et al., 2024; Xu et al., 2024). Note that in PPO (equation 2), contrary to DPO (equation 5), the final model being optimized is not directly aligned using the data $\mathcal{D}$. This is the key intuition behind why PPO-aligned models are less susceptible to privacy threats compared to their DPO counterparts. Next, we state a result about the implicit reward induced by $\pi_{DPO}$.

**Lemma 1.** *Let $\pi_{DPO}$ be the policy obtained by optimizing Equation 5. Then, a) the corresponding implicit rewards on the preference pairs that optimize the BT model loss in Equation 1 are given by $r_d(x, y_w) = \beta \log \frac{\pi_{DPO}(y_w|x)}{\pi_{ref}(y_w|x)}$ and $r_d(x, y_l) = \beta \log \frac{\pi_{DPO}(y_l|x)}{\pi_{ref}(y_l|x)}$ and b) $\pi_{DPO}$ maximizes the objective $\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(\{y_l,y_w\}|x)}[r_d(x, y)] - \beta\mathbb{D}_{KL}[\pi_\theta(y|x)||\pi_{ref}(y|x)]$.*

*Proof.* Claim a) directly follows by subsitiuting the values of $r_d(x, y_w)$ and $r_d(x, y_l)$ in Equation 1. Proof of Claim b) exactly follows the same steps as that of how Equation 3 is the optimal policy for the Objective 2 - given in Rafailov et al. (2024). □

### 2.2 Membership Inference Attacks (MIA) on LLMs

MIA poses a significant privacy risk in the context of LLMs, challenging the security of data used in training such models (Shokri et al., 2017; Nasr et al., 2018). In LLMs, MIAs seek to determine whether specific data was part of the model's training set, exploiting the model's behavior or output nuances to infer data membership. These attacks are particularly concerning for models trained on vast datasets, where inadvertently revealing individual data points could lead to privacy breaches.

The effectiveness of an MIA against LLMs is quantified by a score function $\mathcal{M}$, mapping input samples and the level of access to a real-valued score indicating the likelihood of membership. For a given threshold $\tau$, an input $x$ is classified as a training set member if $\mathcal{M}(x, \text{Access}(\theta)) \geq \tau$.

$$\mathcal{M} : \mathcal{X} \times \text{Access}(\Theta) \to \mathbb{R}. \quad (6)$$

Broadly speaking, MIAs on LLMs can be divided into two categories - Reference-based attacks and Reference-free attacks (Fu et al., 2023). The effectiveness of Reference-based attacks relies on the assumption that training records exhibit a higher probability of being sampled compared to non-training records. By comparing the output probabilities of the target model against those of the reference model, attackers can infer membership status more reliably. Reference-free attacks don't hinge on any such above mentioned assumption and solely rely on the model's tendency to overfit, if any. Thus, they are less effective compared to their Reference-based counterparts. Mattern et. al. (2023) propose a reference-free MIA known as the Neighbour attack specifically for finetuned LLMs; this approach relies on generating synthetic semantically equivalent samples known as Neighbours, given a target point from the finetuning corpus and comparing the model score of the neighboring points to that of the target point. The intuition is that if the model score of the target point is similar to that of the crafted neighbours, then the target point, along with the crafted neighbors are plausible points from the same distribution and the target point is not part of the training set. However, if the model score of the target point is significantly higher compared to the crafted neighbors, then it is more likely that the target point is part of the finetuning data. Zhang et. al. (2024) propose a reference-free MIA known as Divergence-based Calibration Method (DC-PDD) for detecting MIA vulnerabilities for pretraining data of LLMs. DC-PDD relies on publicly available pretraining corpus to construct an within-document token frequency distribution and comparing its divergence against the token probability distribution to derive a detection score. Xie et. al. (2024) propose a reference-free MIA known as Relative Conditional Log Likelihood (ReCALL) for detecting MIA vulnerabilities associated with pretraining data. The idea is to prefix non-member data, which can be obtained using recent data beyond the model training cut-off or generated synthetically, to target points and showing a differential effect on member texts from pretrained corpus vs non-member texts. Note that none of the existing MIAs are designed to factor in the tuple structure of preference data. Further more, the MIAs designed for pretraining data typically have not been effective as the volume of pretraining data is orders of magnitude higher compared to finetuning or RLHF alignment. Duan et al. (2024) conducted an exhaustive experimentation of various MIA attacks (both reference-based and free) on different pretrained LLMs, and observed that most MIAs barely outperform random guessing. This was primarily attributed to two main reasons: a) the large volume of natural language text on which LLMs are pretrained makes the member domains indistinguishable to attacks and b) near-one epoch training prevents over-memorization of pretraining data. Maini et al. (2024) also acknowledge a similar issue with respect to existing MIAs on pretrained LLMs and propose novel complementary attack framework known as Dataset Inference Attack. Research on MIAs targeting LLMs underscores the need for robust privacy-preserving techniques to safeguard training data, with implications for the development and deployment of secure, trustworthy AI systems (Carlini et al., 2020).

### 2.3 Problem Statement

Current research on MIAs has advanced understanding of risks in pre-trained text models, but gaps remain in applying MIAs to preference datasets in LLM alignment. This oversight poses substantial privacy risks, given the critical role of preference data in shaping LLM outputs.

Let $\mathcal{D} = \{(x_i, y_{wi}, y_{li})\}_{i=1}^N$ represent the preference dataset, where $x_i$ is a prompt, $y_{wi}$ is the preferred response, and $y_{li}$ is the less preferred response. The vulnerability of this preference data to MIAs requires a nuanced examination, which can be categorized into three distinct attack vectors:

- **Attack against prompts and individual responses:** This attack determines whether a specific pair of prompt $x$ and response $y_w$ or $y_l$has been used in training, highlighting potential privacy breaches if such data can be identified:

- **Attack against the entire preference tuple:** This more comprehensive attack assesses whether the entire tuple $(x, y_w, y_l)$ can be traced back to the preference set. Note that traditional attack frameworks for SFT or pre-training have only focused on prompts and responses, and have not examined this privacy risk associated with tuples.

This detailed breakdown elucidates the complex vulnerabilities associated with preference data in LLMs. By identifying these specific attack vectors, we aim to advance privacy-preserving mechanisms that safeguard the alignment process and ensure that models respect and protect individual privacy while adhering to human ethical standards.

### 2.4 Theoretical Results

In this section, using analysis similar to that of Li et al. (2023), we first show that DPO has a tendency to overfit on the preference dataset, compared to PPO. Using these results, we comment on the bayes optimal membership inference for both DPO and PPO. We defer all the proofs to Appendix B.

Let $\mathcal{D}_{\text{pair}} = \{(x_i, y_i)\}_{i=1}^n$ be the prompt-response dataset where each $(x_i, y_i)$ is a data pair with $x_i$ being a prompt sampled from the distribution $\mathcal{X}$ and $y_i$ being a response sampled from the conditional distribution $\pi(.|x_i)$. Note that it is possible to construct the preference dataset $\mathcal{D}$ from this notation by repeating the same $x_i$ for both $y_{i_w}$ and $y_{i_l}$. Let $r(x, y)$ be the true reward for any data pair $(x, y)$ and let $\pi_r^*$ be the policy that maximizes the expected reward i.e.

$$\pi_r^\star \leftarrow \arg\max_\pi \mathbb{E}_{x\sim\mathcal{X}}\mathbb{E}_{y\sim\pi(.|x)}[r(x, y)].$$

In practice, we don't know the true reward model $r$. Neither can we compute the true expectations with respect to $\mathcal{X}$ and $\pi(.|x)$. Let $\hat{r}_p$ and $\pi_{PPO}$ be the explicit reward and explicit policy estimated through PPO. Let $\hat{r}_d$ and $\pi_{DPO}$ be the implicit reward and explicit policy estimated through DPO. For ease of exposition, we ignore the regularization terms i.e. we assume $\beta = 0$. Then the PPO and DPO objectives can be stated as follows Li et al. (2023):

$$\pi_{PPO} \leftarrow \underset{\pi}{\text{argmax}} \sum_{i=1}^n \mathbb{E}_{y\sim\pi(.|x_i)} \widehat{r}_p (x_i, y) \tag{7}$$

$$\pi_{DPO} \leftarrow \underset{\pi}{\text{argmax}} \sum_{i=1}^n \widehat{\mathbb{E}}_{y\sim\hat{\pi}(.|x_i)} \widehat{r}_d (x_i, y) \tag{8}$$

where, $\hat{\mathbb{E}}$ denotes the finite sample expectation of the response given a prompt.

*Remark* 1. Equation 7 follows from Equation 5 and Equation 8 follows from claim b) of Lemma 1. Note that it is possible that the learned reward model could err for some $x$ i.e. $\hat{r}(x, y_l) > \hat{r}(x, y_w)$

Following Li et al. (2023), we define three types of error:

1. reward estimation error defined as $\varepsilon_r := \sup_{\hat{r}(x,y)} |\widehat{r}(x, y) - r(x, y)|$,
2. prompt estimation error as $\varepsilon_x := \left|\mathbb{E}_x \mathbb{E}_{y\sim\pi(\cdot|x)}[r(x, y)] - \hat{\mathbb{E}}_x \mathbb{E}_{y\sim\pi(\cdot|x)}[r(x, y)]\right|$, and
3. the response distribution estimation error $\varepsilon_y := \sup_{\pi,r} \sup_x \left|\mathbb{E}_{y\sim\pi(\cdot|x)}[r(x, y)] - \hat{\mathbb{E}}_{y\sim\pi(\cdot|x)}[r(x, y)]\right|$

Here, $\sup_{\pi,r} := \sup_\pi \sup_{r:\mathcal{X}\times\mathcal{Y}}$; and $\hat{\mathbb{E}}_x$ and $\hat{\mathbb{E}}_y$ denote the finite-sample estimations of expectation under the preference dataset's prompt and response distributions, respectively.

Let $\pi_r^*$ be the optimal policy which maximizes the true reward over the preference dataset i.e. $\pi_r^* := \arg\max_\pi \hat{\mathbb{E}}_x \hat{\mathbb{E}}_{y\sim\pi}[r(x, y)]$. Let $r(\pi)$ be the finite sample expectation true reward for any policy $\pi$ i.e. $r(\pi) := \hat{\mathbb{E}}_x \hat{\mathbb{E}}_{y\sim\pi}[r(x, y)]$

**Proposition 1.** $r(\pi_r^*) - r(\pi_{PPO}) \leq 2\varepsilon_r + 2\varepsilon_x$ *;* $r(\pi_r^*) - r(\pi_{DPO}) \leq 2\varepsilon_r$

If $\pi_r^*$ and $r(\pi)$ were defined on the population rather than the preference dataset i.e. $\pi_r^* := \arg\max_\pi \mathbb{E}_x \mathbb{E}_{y\sim\pi}[r(x, y)]$ and $r(\pi) := \mathbb{E}_x \mathbb{E}_{y\sim\pi}[r(x, y)]$ , then from Li et al. (2023), we have

**Proposition 2.** $r(\pi_r^*) - r(\pi_{PPO}) \leq 2\varepsilon_r + 2\varepsilon_x$ *;* $r(\pi_r^*) - r(\pi_{DPO}) \leq 2\varepsilon_r + 2\varepsilon_x + 2\varepsilon_y$

From proposition 1 and 2, we can interpret that DPO overfits on the preference dataset used for alignment and results in poor generalization, as compared to PPO. Next, we use results from Sablayrolles et al. (2019); Aubinais et al. (2023) to lower-bound the bayes optimal membership when an algorithm is overfitting on the training dataset.

To this end, we consider a general learning model $\pi$ that produces the parameters $\theta_n$ minimizing the loss $L_n$ :

$\theta \rightarrow \frac{1}{n}\sum_{i=1}^{n} l_\theta(x_i, y_i)$ for some training dataset $\{z_i := (x_i, y_i)\}_{i=1}^{n}$, where $z_i$ are independent and identically sampled from the same population. Similar to Sablayrolles et al. (2019), we assume the posterior distribution of $\theta_n$ given $\{z_i\}_{i=1}^{N}$ follows:

$$P(\theta_n|z_1, \dots, z_n) \propto e^{-\frac{1}{T}\sum_{i=1}^{n}\ell(\theta_n, z_i)} \tag{9}$$

where $T$ is a temperature parameter to factor in various randomness present in training algorithms such as sampling, dropout, etc (Sablayrolles et al., 2019; Aubinais et al., 2023). Given model parameters $\theta_n$ obtained using an algorithm $\pi$, we wish to know how much information is contained in $\theta_n$ about the memberships $m_i \in \{0, 1\}$ corresponding to $z_i$ for $i \in \{1, \cdots, n\}$. Without loss of generality, let $z_1 = (x_1, y_1)$ be a prompt response pair whose membership we want to infer from an aligned model $\pi_\theta$. We want to determine the membership of $z_1$ i.e. $\mathcal{M}(\pi_\theta, z_1) := P(m_1 = 1|\pi_\theta, z_1)$. We collect information about the remaining data as $\mathcal{T} = \{z_2, \dots, z_n\}$

We state a modified version of Theorem 2 of Sablayrolles et al. (2019) as a proposition here. For brevity, we will drop the subscript in $\theta_n$ when the context is clear.

**Lemma 2.** *Given the parameters $\theta$ and sample $z_1$, assuming a uniform prior over $m_1$'s, the optimal membership inference is given by:*

$$\mathcal{M}(\theta, z_1) = \mathbb{E}_{\mathcal{T}}[\sigma(s(z_1, \theta, p_{\mathcal{T}}))] \tag{10}$$

*where we define the corresponding posterior distribution $p_{\mathcal{T}}(\theta)$, threshold $\tau_p$, and the score $s$ as follows:*

$$p_{\mathcal{T}}(\theta) := \frac{e^{-\frac{1}{T}\sum_{i=1}^{n}\ell(\theta, z_i)}}{\int_t e^{-\frac{1}{T}\sum_{i=1}^{n}\ell(t, z_i)}\,dt} \tag{11}$$

$$\tau_p(z_1) := T\log\left(\int_t e^{\frac{1}{T}\ell(t, z_1)} p_{\mathcal{T}}(t)dt\right) \tag{12}$$

$$s(z_1, \theta, p_{\mathcal{T}}) := \frac{1}{T}\left(\tau_p(z_1) - \ell(\theta, z_1)\right) \tag{13}$$

The interpretation of Equation 10 is as follows: the lower the loss $\ell(\theta, z_1)$ becomes compared to the threshold $\tau_p(z_1)$, the more we gain non-trivial information about the membership of $z_1$. Here $\sigma$ denotes the sigmoid function. In general, the posterior $p_{\mathcal{T}}$ is intractable.

The MALT (acronym for Membership Attack Loss Threshold) assumption is a simplification framework to handle the intractability of posterior of $p_{\mathcal{T}}(\theta)$ and it assumes that the posterior $\tau_p := \log\left(\int_t e^{-\frac{1}{T}\ell(t, z_1)} p(t)dt\right)$ is constant (Sablayrolles et al., 2019). Despite the simplistic assumption, it has been show that MALT has decent empirical performance (Sablayrolles et al., 2019; Carlini et al., 2019; Quan et al., 2022). Under this MALT assumption, it is possible to show that for the setting considered in Equations 7 and 8, $M(\pi_{DPO}, z_1) \geq M(\pi_{PPO}, z_1)$.

**Proposition 3.** *If $\pi_{PPO}$ and $\pi_{DPO}$ are the global optimisers of the PPO and DPO objectives in Equation 7 and 8 respectively, then $M(\pi_{DPO}, z_1) \geq M(\pi_{PPO}, z_1)$ under the MALT assumption of Sablayrolles et al. (2019).*

In general, MALT assumption is quite restrictive in nature. Hence, we don't restrict ourselves to MALT assumption and give an generalized fundamental lower bound on the score function of the optimal membership for any algorithm that overfits the training data.

Next we consider the case where the learning model $\pi$ overfits on the training data. Following Aubinais et al. (2023), we first define a probabilistic version of overfitting as follows:

**Definition 1.** An algorithm $\pi$ is $(\varepsilon, 1-\alpha)$ overfitting for some $\varepsilon \in \mathbb{R}^+$ and $\alpha \in (0, 1)$ when $\mathbb{P}\left(\ell_\theta(z_1) \leq \varepsilon\right) \geq 1 - \alpha$.

Next, we make a simplifying assumption that under the posterior distribution given in 9, the likelihoods $\{\ell_\theta(z_i)\}$ are independent and identically distributed. Using this simplifying assumption, we give a sufficient condition for the above defined probabilistic overfitting based on the stopping criterion for the algorithm $\pi$.

**Lemma 3.** *For some fixed $\varepsilon \in \mathbb{R}^+$ and $\alpha \in (0, 1)$, let $\eta := \varepsilon\alpha$, and suppose that $\pi$ stops as soon as $L_n \leq \eta$, then $\pi$ is $\left(\varepsilon, \left(1 - \left(1 - \frac{1}{n\alpha}\right)^n\right)\right)$ overfitting.*

Note that the probability of overfitting given in Lemma 3 i.e. $\left(1 - \left(1 - \frac{1}{n\alpha}\right)^n\right)$ is much higher compared to the $1 - \alpha$ mentioned in proposition 4.2 of Aubinais et al. (2023).

**Theorem 2.1.** *For some fixed $\varepsilon$ and $\alpha \in (0, 1)$, let $\eta := \varepsilon\alpha$ and the algorithm $\pi$ stops when $L_n(\hat{\theta}) \leq \eta$, then*

$$s(z_1, \theta, p_{\mathcal{T}}) \geq \log\left(1 - \left(1 - \frac{1}{n\alpha}\right)^n\right) - \frac{\ell(\theta, z_1)}{T} \tag{14}$$

There are two practical conclusions from Theorem 2.1 as follows: (a) keeping everything else constant, as $\alpha \to 0$ , the score function increases i.e. more the overfitting, more the susceptibility towards MIA. (b) as the number of training points $n$ increases, the score function decreases. That is, smaller the dataset size, higher the privacy risk.

From Propositions 1 and 2, we see that DPO is prone to overfitting, and thus, by Theorem 2.1, it has higher susceptibility to MIA compared to PPO. In §3, we introduce our novel framework and in §4, we empirically show that indeed DPO has serious MIA vulnerabilities as compared to PPO.

## 3 PREMIA FRAMEWORK

There are two major motivations for proposing a novel optimstic attack framework: a) Traditional MIAs fail to account for the unique characteristics of preference data and the specific optimization objectives employed during the alignment of large language models (LLMs). These frameworks primarily focus on assessing attacks at the prompt-response level, neglecting the nuanced structure of preference tuples—a widely adopted data format for LLM alignment. Consequently, there is a pressing need for tailored frameworks capable of evaluating MIA not only at the prompt-response level but also at the granular preference tuple level, capturing the true extant of MIA in LLM alignment.

b) Most of the attack frameworks proposed in literature are reference-free in nature, which struggle to achieve effective results when applied to LLMs which have remarkable generalization capabilities at the distributional level. However, under the optimistic assumption of access to the base model employed for fine-tuning, it becomes feasible to design reference-based attacks, known for their superior effectiveness. Leveraging this assumption, our proposed PREMIA framework carefully crafts a reference-based attack tailored to the intricacies of DPO and PPO — two of the widely used alignment techniques for LLMs.

The assumption of base model access, while optimistic, is not impractical in the context of LLM alignment. Many non-enterprise organizations and researchers opt to fine-tune and align open-source LLMs for their specific tasks, as closed-source models like Claude or ChatGPT often come with restrictive licenses and relatively higher inference costs. Moreover, certain government and institutional regulations prohibit the transfer of data outside specific premises, leaving local fine-tuning, alignment, and on-premise hosting of LLMs as the only viable option. By leveraging these open-source base models, practitioners can tailor the LLMs to their requirements while avoiding the limitations and expenses associated with proprietary models. Consequently, the assumption of base model access is neither entirely impractical nor over-convenient, making our proposed PREMIA framework applicable to a wide range of real-world scenarios. Further, since reference-based attacks tend to perform relatively better at detection of membership, PREMIA, with its optimistic assumption gives a practical upper bound on the extent of vulnerability.

### 3.1 For Individual Response

For individual prompt-response pairs, we define the metric as the ratio of the conditional probability of the aligned model $\pi_\theta$ to that of a reference model $\pi_{\text{ref}}$:

$$\rho_y = \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}, \tag{15}$$

This ratio measures the likelihood that the target model will produce a specific response compared to the reference model, indicating potential overfitting to training data. The motivation behind this specific functional form is in part due to the functional form of the reward being optimized in both PPO and DPO - refer to Equation 4.

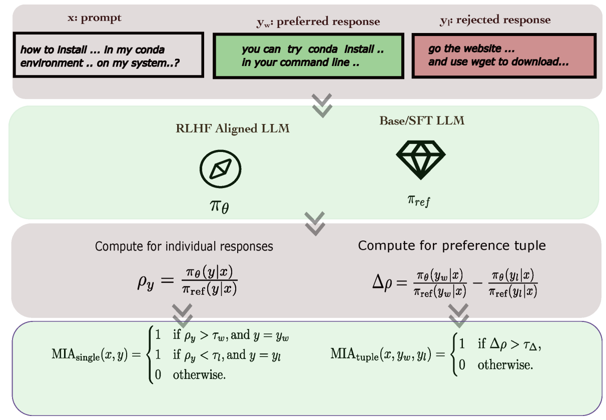


Figure 1: Overview of PREMIA framework for individual prompt-response pairs and the entire preference tuple

Building on top of the above metric, depending on whether the response is chosen or rejected, we define the pair's membership as:

$$\text{MIA}_{\text{single}}(x, y) = \begin{cases} 1 & \text{if } \rho_y > \tau_w, \text{and } y = y_w \\ 1 & \text{if } \rho_y < \tau_l, \text{and } y = y_l \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

Although $\tau_y$ is mentioned, our primary metric in the experiments is the Area Under the Receiver Operating Characteristic (AUROC), which does not require setting a specific threshold.

The choice of the reference model $\pi_{\text{ref}}$ serves as a benchmark for comparing the behavior of the target model $\pi_\theta$. This model can be the base pre-trained model from which $\pi_\theta$ originated or a different base model trained on the same dataset. Our experiments, designed to test both scenarios, consistently demonstrate robust performance of our MIA method under various conditions.

### 3.2 For the Entire Preference Tuple

To ascertain the membership of the complete preference tuple $(x, y_w, y_l)$, we compute the difference between the probability ratios of the preferred and not preferred responses:

$$\Delta\rho = \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}. \tag{17}$$

This measure captures the comparative preference strength more effectively, offering a nuanced insight into how preference data impacts model training:

$$\text{MIA}_{\text{tuple}}(x, y_w, y_l) = \begin{cases} 1 & \text{if } \Delta\rho > \tau_\Delta, \\ 0 & \text{otherwise.} \end{cases} \tag{18}$$

# 4 EXPERIMENTS

## 4.1 Research Questions

This subsection outlines the practical research questions guiding our experimental design. More specifically we study: a) *How do DPO and PPO differ in their susceptibility to Membership Inference Attacks?* b) *Does model size influence its risk of data leakage through MIAs, and how does this vary between DPO and PPO trained models?* and c) *What are the performance and privacy trade-offs when employing DPO versus PPO in LLMs?*. These questions aim to evaluate the comparative effectiveness, privacy implications, and utility of DPO and PPO in aligning LLMs.

## 4.2 Setup

**Models.** We conduct experiments using a variety of models to ensure a comprehensive evaluation on different scales of model complexity. We include larger models such as Gemma-2-2B (Gemma, 2024), Mistral-7B-v0.3, Mistral-7B-v0.1 (Jiang et al., 2023), Open-llama-3b and Open-llama-7b models (Geng and Liu, 2023; Touvron et al., 2023) as well as a series of smaller models from the OpenAI GPT-2 family (Radford et al., 2019): GPT2, GPT2-medium, GPT2-large, and GPT2-xl. For the reference model in our ratio calculations, we primarily use the SFT model trained from the same base pre-trained version of the model being evaluated. Additionally, we conduct experiments where the reference model differs from the base model to evaluate the robustness of our methodology under varied conditions.

**Datasets.** For our experiments, we utilize the Stack-Exchange-Paired (SE) dataset[*] and the IMDB-RLHF-Pair dataset (Rafailov et al., 2024). Both datasets have a prompt $x$ accompanied by two responses: the 'Chosen' response $y_w$ and the 'Rejected' response $y_l$. The SE dataset contains questions and answers from the Stack Overflow dataset, where answers with more votes are preferred. The IMDB-RLHF-Pair dataset is generated by IMDB, and responses with positive sentiment are preferred. For the Stack-Exchange-Paired dataset, the `data/rl` split is used for training, and `data/evaluation` is used as validation data. For the IMDB-RLHF-Pair dataset, 20k entries are used for training, while the remaining is for validation.

**Evaluation Metrics.** To comprehensively assess DPO and PPO alignment, we employ a dual-focused evaluation framework encompassing utility performance and membership privacy:

- *Utility Performance:* Our evaluation includes the reward score of generated responses given by the reward model and perplexity for assessing fluency. We also incorporate comprehensive diversity measures: Mean Segmented Type Token Ratio (MSSTR), Distinct-1, Distinct-2, Unique-1, and Unique-2 metrics (Johnson, 1944; Li et al., 2015; Ramamurthy et al., 2022). Additionally, we utilize advanced text generation quality metrics such as BERTScore Zhang et al. (2019), ROUGE Lin (2004), BLEU Papineni et al. (2002), and METEOR Banerjee and Lavie (2005), which collectively offer a nuanced view of the models' performance in terms of fluency, adequacy, and diversity, closely mirroring human judgment in text quality assessment.
- *MIA Performance:* In line with previous literature on MIA (Shi et al., 2024; Zhang et al., 2024; Chen et al., 2022), to measure the model's susceptibility, we utilize the Area Under the Receiver Operating Characteristic curve (AUROC). This metric captures the model's defense against MIAs, reflecting the balance between true positive rate and false positive rate in identifying training data.

**Implementation Details.** Due to the computational efficiency of LoRA, we used LoRA for all of our model training processes. Additionally, we hypothesized that fine-tuning LoRA at the RL stage would help to ensure that the aligned model does not deviate significantly from the reference model. To further improve efficiency, we also used quantization techniques. We use TRL(von Werra et al., 2023) for model alignment training. More detailed implementation information can be found in Appendix C.

## 4.3 Existing MIA frameworks

To accurately evaluate the differential susceptibility, in addition to the optimistic PREMIA, we also compare using well-known MIA frameworks such Perplexity(Yeom et al., 2018), Zlib(Carlini et al., 2021), Lowercase(Carlini et al., 2021), Min-k(Shi et al., 2023), etc. which are specifically tailored for LLMs. More details about these baselines are given in Appendix A. These frameworks are designed to target individual prompt-response pairs of the preference data but do not extend to analyzing the entire preference tuples.

### 4.3.1 Differential Susceptibility - DPO vs PPO

Table 1 presents the AUROC scores for the 'Chosen' and 'Rejected' responses for various MIA methods across Mistral-7B, Open-llama-3b, and Open-llama-7b models. PREMIA-base and PREMIA-SFT indicate using the base model or SFT model as the reference model respectively. In general, we find that, compared to PPO, DPO is flagged to be highly susceptible for both the datasets across all the frameworks. As expected, PREMIA being an optimistic framework consistently achieves high AUROC scores. We

[*] `https://huggingface.co/datasets/lvwerra/stack-exchange-paired`

Figure 2: AUROC scores for $\text{MIA}_{\text{Pair}}$ detection for SE dataset

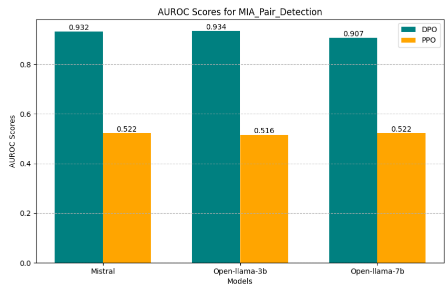
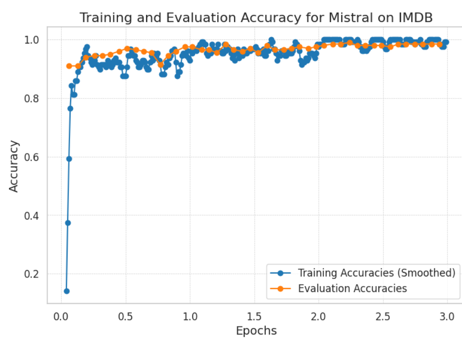



Figure 3: Train/Eval Aaccuracy for Mistral-7B on IMDB.




do not measure the entire tuple using baselines because traditional MIA methods are not designed to handle the contextual dependencies inherent in preference tuples. AUROC numbers for PREMIA-SFT on the tuple memberships for SE dataset is given in Figure 2; Appendix D.3 has the corresponding plot for IMDB dataset. In order to evaluate the robustness of PREMIA, we switch the reference models within the same family and evaluate the AUROC. As shown in Table 3, this did not deteriorate the performance significantly.

### 4.3.2 Impact of Model Size on MIA Effectiveness

Table 4 details the PREMIA-SFT results for models of different sizes in the GPT series on the SE and IMDB datasets; refer Appendix D.1 for results from other frameworks. First, we note that in the GPT2 models as well DPO demonstrates a higher vulnerability compared to PPO. Looking at the AUROC values in both Table 1 and 4 we note that PPO aligned models are nearly impregnable to MIA with all of the existing frameworks. Second, we observe that, when compared to Mistral-7B and OpenLlama models, GPT2 models aligned using DPO show higher vulnerability on both the tasks. When compared between the tasks, we note that vulnerability is higher for SE dataset compared to IMDB dataset. This is because IMDB task is relatively easier (wherein the model has has to generate positive responses), compared to SE task (the model has to produce an answer for an SE question). As shown in Fig. 3, Mistral-7B

Table 1: AUROC scores comparing different MIA methods on Gemma-2-2B, Mistral-7B-v{0.1,0.3}, Open-llama-3b, and Open-llama-7b models are presented, where higher scores indicate greater susceptibility to MIA. The best and second-best scores in each column are highlighted in orange and green, respectively. The better score between DPO and PPO trained models is underlined.

| | | IMDB | | | | Stack-Exchange | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | $\text{MIA}_{\text{Chosen}}$ | | $\text{MIA}_{\text{Rejected}}$ | | $\text{MIA}_{\text{Chosen}}$ | | $\text{MIA}_{\text{Rejected}}$ | |
| | | DPO | PPO | DPO | PPO | DPO | PPO | DPO | PPO |
| Gemma-2-2B | PPL | 0.572 | 0.531 | 0.575 | 0.541 | 0.557 | 0.505 | 0.542 | 0.517 |
| | Zlib | 0.589 | 0.532 | 0.591 | 0.531 | 0.579 | 0.517 | 0.589 | 0.534 |
| | Lowercase | 0.556 | 0.538 | 0.545 | 0.525 | 0.565 | 0.531 | 0.556 | 0.547 |
| | Ref | 0.570 | 0.530 | 0.576 | 0.547 | 0.552 | 0.527 | 0.570 | 0.536 |
| | MIN-K | 0.571 | 0.524 | 0.587 | 0.523 | 0.605 | 0.538 | 0.571 | 0.551 |
| | N-hood | 0.575 | 0.512 | 0.568 | 0.513 | 0.582 | 0.519 | 0.632 | 0.521 |
| | PREMIA-base | 0.581 | 0.539 | 0.585 | 0.555 | 0.691 | 0.534 | 0.671 | 0.545 |
| | PREMIA-SFT | 0.592 | 0.545 | 0.597 | 0.552 | 0.714 | 0.539 | 0.681 | 0.547 |
| Mistral-7B-v0.3 | PPL | 0.592 | 0.525 | 0.592 | 0.525 | 0.575 | 0.521 | 0.592 | 0.525 |
| | Zlib | 0.598 | 0.521 | 0.598 | 0.521 | 0.584 | 0.529 | 0.598 | 0.521 |
| | Lowercase | 0.567 | 0.505 | 0.567 | 0.505 | 0.581 | 0.514 | 0.567 | 0.505 |
| | Ref | 0.587 | 0.538 | 0.587 | 0.538 | 0.572 | 0.523 | 0.587 | 0.538 |
| | MIN-K | 0.589 | 0.534 | 0.589 | 0.534 | 0.578 | 0.508 | 0.589 | 0.534 |
| | N-hood | 0.576 | 0.509 | 0.577 | 0.512 | 0.582 | 0.519 | 0.632 | 0.514 |
| | PREMIA-base | 0.596 | 0.543 | 0.596 | 0.543 | 0.758 | 0.531 | 0.596 | 0.543 |
| | PREMIA-SFT | 0.619 | 0.558 | 0.619 | 0.558 | 0.789 | 0.543 | 0.619 | 0.558 |
| Mistral-7B-v0.1 | PPL | 0.569 | 0.538 | 0.588 | 0.503 | 0.575 | 0.512 | 0.586 | 0.515 |
| | Zlib | 0.593 | 0.568 | 0.606 | 0.536 | 0.539 | 0.529 | 0.553 | 0.516 |
| | Lowercase | 0.516 | 0.509 | 0.515 | 0.501 | 0.571 | 0.523 | 0.574 | 0.520 |
| | Ref | 0.571 | 0.533 | 0.607 | 0.511 | 0.572 | 0.518 | 0.589 | 0.512 |
| | MIN-K | 0.564 | 0.511 | 0.582 | 0.509 | 0.582 | 0.519 | 0.632 | 0.514 |
| | N-hood | 0.548 | 0.504 | 0.528 | 0.516 | 0.600 | 0.513 | 0.568 | 0.506 |
| | PREMIA-base | 0.571 | 0.523 | 0.592 | 0.517 | 0.771 | 0.523 | 0.764 | 0.526 |
| | PREMIA-SFT | 0.576 | 0.507 | 0.590 | 0.527 | 0.803 | 0.521 | 0.760 | 0.520 |
| Open-llama-3b | PPL | 0.580 | 0.508 | 0.573 | 0.526 | 0.574 | 0.515 | 0.544 | 0.512 |
| | Zlib | 0.602 | 0.540 | 0.595 | 0.506 | 0.539 | 0.517 | 0.514 | 0.521 |
| | Lowercase | 0.541 | 0.556 | 0.546 | 0.551 | 0.633 | 0.517 | 0.577 | 0.523 |
| | Ref | 0.587 | 0.508 | 0.590 | 0.535 | 0.596 | 0.524 | 0.582 | 0.518 |
| | MIN-K | 0.587 | 0.523 | 0.579 | 0.529 | 0.578 | 0.524 | 0.599 | 0.516 |
| | N-hood | 0.567 | 0.511 | 0.586 | 0.521 | 0.601 | 0.523 | 0.583 | 0.520 |
| | PREMIA-base | 0.564 | 0.520 | 0.562 | 0.540 | 0.758 | 0.531 | 0.750 | 0.525 |
| | PREMIA-SFT | 0.594 | 0.504 | 0.609 | 0.518 | 0.789 | 0.525 | 0.761 | 0.534 |
| Open-llama-7b | PPL | 0.577 | 0.529 | 0.572 | 0.505 | 0.577 | 0.514 | 0.551 | 0.527 |
| | Zlib | 0.599 | 0.559 | 0.593 | 0.525 | 0.535 | 0.509 | 0.531 | 0.516 |
| | Lowercase | 0.537 | 0.501 | 0.540 | 0.502 | 0.548 | 0.516 | 0.569 | 0.518 |
| | Ref | 0.583 | 0.515 | 0.586 | 0.503 | 0.605 | 0.516 | 0.580 | 0.517 |
| | MIN-K | 0.597 | 0.527 | 0.583 | 0.511 | 0.567 | 0.510 | 0.590 | 0.530 |
| | N-hood | 0.563 | 0.507 | 0.576 | 0.517 | 0.598 | 0.525 | 0.576 | 0.514 |
| | PREMIA-base | 0.559 | 0.511 | 0.560 | 0.504 | 0.761 | 0.523 | 0.759 | 0.516 |
| | PREMIA-SFT | 0.594 | 0.511 | 0.611 | 0.527 | 0.774 | 0.516 | 0.730 | 0.524 |

Table 2: Privacy vs Utility Trade-off analysis on the Mistral-7B model on SE dataset.

| | Base | SFT | PPO | DPO |
|---|---|---|---|---|
| $MIA_{Chosen}$ | — | 0.53 | 0.52 | 0.80 |
| $MIA_{Rejected}$ | — | 0.61 | 0.52 | 0.76 |
| $MIA_{Pair}$ | — | 0.55 | 0.52 | 0.93 |
| Reward↑ | -1.922 | -1.953 | -0.771 | -1.035 |
| PPL↓ | 11.148 | 7.673 | 11.671 | 14.991 |
| msttr-100↑ | 0.673 | 0.651 | 0.633 | 0.640 |
| distinct 1↑ | 0.180 | 0.127 | 0.085 | 0.123 |
| distinct 2↑ | 0.631 | 0.521 | 0.422 | 0.520 |
| unique 1↑ | 2010 | 3213 | 3530 | 3059 |
| unique 2↑ | 9507 | 17238 | 25205 | 18017 |
| Bert Score↑ | 0.876 | 0.879 | 0.883 | 0.877 |
| ROUGE↑ | 0.424 | 0.458 | 0.457 | 0.443 |
| BLEU↑ | 0.348 | 0.367 | 0.338 | 0.360 |
| METEOR↑ | 0.445 | 0.467 | 0.449 | 0.466 |

Table 3: PREMIA of Openllama-7b when Openllama-3b model is used as reference model.

| dataset | $\pi_\theta/\pi_{ref}$ | chosen | rejected | pair |
|---|---|---|---|---|
| SE | open-llama-3b/7b | 0.754 | 0.673 | 0.863 |
| | open-llama-7b/3b | 0.722 | 0.732 | 0.882 |
| IMDB | open-llama-3b/7b | 0.585 | 0.592 | 0.512 |
| | open-llama-7b/3b | 0.581 | 0.590 | 0.508 |

aligned with DPO achieves over 90% accuracy in distinguishing between selected and rejected responses in only the first 0.2 epoch for the IMDB dataset. Large pre-trained models like Mistral-7B already have strong generalization capabilities, which undermines the effectiveness of MIA. Similarly, large GPT2 models such as GPT2-xl show better generalization on simple tasks, making them less susceptible to MIA (Tänzer et al., 2021; Wei et al., 2024). Note that this highlights a new understanding on how MIA varies with model size - on pre-training data, Shi et al. (2024) found that model vulnerability increases with model size for MIA; whereas for preference data, we notice that the relationship is a bit more nuanced, depending on the task complexity during alignment.

#### 4.3.3 Trade-Off between Performance and Privacy

Table 2 analyzes the trade-off between vulnerability to MIA and model utility between DPO and PPO for Mistral-7B on SE dataset. The "Reward" row represents the average reward score given by the reward model for each of these models, indicating how well the task was accomplished. Clearly, DPO and PPO have better reward and utility metrics compared to the rest. Further, DPO is clearly more vulnerable to MIA. It is worth noting that PPO provides similar utility performance to DPO, but it has a lower AUROC. These findings are in line with existing research, which also shows that despite DPO being relatively straightforward to train, it does not improve the model performance compared to PPO Ivison et al. (2024); Xu et al. (2024). Anecdotal responses from both these datasets to various alignment approaches are given in Appendix D.4.

Table 4: Performance of PREMIA-SFT on various GPT2 model variants across SE and IMDB datasets.

| | | $MIA_{Chosen}$ | | $MIA_{Rejected}$ | | $MIA_{Pair}$ | |
|---|---|---|---|---|---|---|---|
| | | DPO | PPO | DPO | PPO | DPO | PPO |
| Stack Exchange | GPT2 | 0.824 | 0.513 | 0.749 | 0.510 | 0.899 | 0.515 |
| | GPT2-medium | 0.829 | 0.518 | 0.752 | 0.530 | 0.910 | 0.536 |
| | GPT2-large | 0.857 | 0.508 | 0.758 | 0.519 | 0.916 | 0.517 |
| | GPT2-xl | 0.858 | 0.516 | 0.788 | 0.509 | 0.928 | 0.519 |
| | Open-llama-3b | 0.789 | 0.525 | 0.761 | 0.534 | 0.934 | 0.516 |
| | Open-llama-7b | 0.774 | 0.515 | 0.730 | 0.523 | 0.907 | 0.526 |
| | Mistral-7B | 0.803 | 0.520 | 0.760 | 0.520 | 0.932 | 0.522 |
| IMDB | GPT2 | 0.636 | 0.550 | 0.713 | 0.511 | 0.771 | 0.549 |
| | GPT2-medium | 0.641 | 0.549 | 0.707 | 0.539 | 0.762 | 0.528 |
| | GPT2-large | 0.615 | 0.611 | 0.659 | 0.583 | 0.704 | 0.520 |
| | GPT2-xl | 0.623 | 0.591 | 0.643 | 0.579 | 0.692 | 0.519 |
| | Open-llama-3b | 0.594 | 0.503 | 0.609 | 0.518 | 0.509 | 0.517 |
| | Open-llama-7b | 0.594 | 0.510 | 0.611 | 0.526 | 0.500 | 0.512 |
| | Mistral-7B | 0.572 | 0.507 | 0.611 | 0.527 | 0.556 | 0.513 |

## 5 CONCLUSION AND LIMITATIONS

This paper examines the vulnerability of preference datasets in LLM alignment to MIAs. We show both theoretically and empirically that models trained with DPO are more susceptible to MIAs than those using PPO. Further, we also discover that the extent of MIA depends on various factors such as model size, task complexity, etc. While we have focused on highlighting MIA vulnerabilites, we haven't touched upon the various ways to mitigate them. Techniques such as DP-SGD Abadi et al. (2016), model pruning (Han et al., 2015), and privacy-aware losses (Chen et al., 2022) could be evaluated in the context of MIA on preference data. While optimistic frameworks such as PREMIA work for opensource models, there is a need for designing effective frameworks for closed-source LLMs where there is no access to the base model.

### References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. (2016). Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318. ACM.

Aubinais, E., Gassiat, E., and Piantanida, P. (2023). Fundamental limits of membership inference attacks on machine learning models. *arXiv preprint arXiv:2310.13786*.

Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., et al. (2022). Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Banerjee, S. and Lavie, A. (2005). Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Carlini, N. et al. (2021). Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*.

Carlini, N., Liu, C., Erlingsson, Ú., Kos, J., and Song, D. (2019). The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX security symposium (USENIX security 19)*, pages 267–284.

Carlini, N., Tramèr, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T. B., Song, D., Erlingsson, Ú., et al. (2020). Extracting training data from large language models. corr abs/2012.07805 (2020). *arXiv preprint arXiv:2012.07805*.

Chen, D., Yu, N., and Fritz, M. (2022). Relaxloss: Defending membership inference attacks without losing utility. In *International Conference on Learning Representations*.

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. (2017). Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Duan, M., Suri, A., Mireshghallah, N., Min, S., Shi, W., Zettlemoyer, L., Tsvetkov, Y., Choi, Y., Evans, D., and Hajishirzi, H. (2024). Do membership inference attacks work on large language models? *arXiv preprint arXiv:2402.07841*.

Fu, W., Wang, H., Gao, C., Liu, G., Li, Y., and Jiang, T. (2023). Practical membership inference attacks against fine-tuned large language models via self-prompt calibration. *arXiv preprint arXiv:2311.06062*.

Gemma, T. (2024). Gemma.

Geng, X. and Liu, H. (2023). Openllama: An open reproduction of llama.

Han, S., Mao, H., and Dally, W. J. (2015). Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*.

Hendrycks, D., Carlini, N., Schulman, J., and Steinhardt, J. (2021). Unsolved problems in ml safety. *arXiv preprint arXiv:2109.13916*.

Ivison, H., Wang, Y., Liu, J., Wu, Z., Pyatkin, V., Lambert, N., Smith, N. A., Choi, Y., and Hajishirzi, H. (2024). Unpacking dpo and ppo: Disentangling best practices for learning from preference feedback.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. d. l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al. (2023). Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Johnson, W. (1944). Studies in language behavior: A program of research. *Psychological Monographs*, 56(2):1–15.

Li, J., Galley, M., Brockett, C., Gao, J., and Dolan, B. (2015). A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*.

Li, Z., Xu, T., and Yu, Y. (2023). Policy optimization in rlhf: The impact of out-of-preference data. *arXiv preprint arXiv:2312.10584*.

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Maini, P., Jia, H., Papernot, N., and Dziedzic, A. (2024). Llm dataset inference: Did you train on my dataset? *arXiv preprint arXiv:2406.06443*.

Nasr, M., Shokri, R., and Houmansadr, A. (2018). Comprehensive privacy analysis of deep learning. In *Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP)*, volume 2018, pages 1–15.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. (2022). Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Quan, P., Chakraborty, S., Jeyakumar, J. V., and Srivastava, M. (2022). On the amplification of security and privacy risks by post-hoc explanations in machine learning models. *arXiv preprint arXiv:2206.14004*.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. (2024). Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Ramamurthy, R., Ammanabrolu, P., Brantley, K., Hessel, J., Sifa, R., Bauckhage, C., Hajishirzi, H., and Choi, Y. (2022). Is reinforcement learning (not) for natural language processing: Benchmarks, baselines, and building blocks for natural language policy optimization. *arXiv preprint arXiv:2210.01241*.

Sablayrolles, A., Douze, M., Schmid, C., Ollivier, Y., and Jégou, H. (2019). White-box vs black-box: Bayes optimal strategies for membership inference. In *Interna-*

*tional Conference on Machine Learning*, pages 5558–5567. PMLR.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Shi, W., Ajith, A., Xia, M., Huang, Y., Liu, D., Blevins, T., Chen, D., and Zettlemoyer, L. (2023). Detecting pretraining data from large language models. *arXiv preprint arXiv:2310.16789*.

Shi, W., Ajith, A., Xia, M., Huang, Y., Liu, D., Blevins, T., Chen, D., and Zettlemoyer, L. (2024). Detecting pretraining data from large language models. In *The Twelfth International Conference on Learning Representations*.

Shokri, R., Stronati, M., Song, C., and Shmatikov, V. (2017). Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE.

Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. F. (2020). Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021.

Tänzer, M., Ruder, S., and Rei, M. (2021). Memorisation versus generalisation in pre-trained language models. *arXiv preprint arXiv:2105.00828*.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. (2023). Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

von Werra, L., Belkada, Y., Tunstall, L., Beeching, E., Thrush, T., Lambert, N., Huang, S., Rasul, K., and Gallouédec, Q. (2023). Trl: Transformer reinforcement learning.

Wei, J., Zhang, Y., Zhang, L. Y., Ding, M., Chen, C., Ong, K.-L., Zhang, J., and Xiang, Y. (2024). Memorization in deep learning: A survey. *arXiv preprint arXiv:2406.03880*.

Xu, S., Fu, W., Gao, J., Ye, W., Liu, W., Mei, Z., Wang, G., Yu, C., and Wu, Y. (2024). Is dpo superior to ppo for llm alignment? a comprehensive study. *arXiv preprint arXiv:2404.10719*.

Yang, J., Jin, H., Tang, R., Han, X., Feng, Q., Jiang, H., Zhong, S., Yin, B., and Hu, X. (2024). Harnessing the power of llms in practice: A survey on chatgpt and beyond. *ACM Transactions on Knowledge Discovery from Data*, 18(6):1–32.

Yeom, S. et al. (2018). Privacy risk in machine learning: Analyzing the connection to overfitting. *IEEE 31st Computer Security Foundations Symposium (CSF)*.

Zhang, J., Sun, J., Yeats, E., Ouyang, Y., Kuo, M., Zhang, J., Yang, H., and Li, H. (2024). Min-k%++: Improved baseline for detecting pre-training data from large language models. *arXiv preprint arXiv:2404.02936*.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2019). Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., et al. (2023). A survey of large language models. *arXiv preprint arXiv:2303.18223*.

## Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Not Applicable]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes] - as footnotes in §4.2
2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes] - in Appendix B
   (c) Clear explanations of any assumptions. [Yes]
3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Not Applicable]

   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

# Instructions for Paper Submissions to AISTATS 2025: Supplementary Materials

# Appendix

## Table of Contents



## A DETAILS OF BASELINES

**Perplexity (PPL):** The loss attack method, based on the approach outlined in Yeom et al. (2018), utilizes the perplexity of a sequence to gauge how well a language model predicts the tokens within that sequence. Perplexity is defined as:

$$\mathcal{P} = \exp\left(-\frac{1}{n}\sum_{i=1}^{n}\log\pi_\theta(x_i|x_1,\ldots,x_{i-1})\right), \tag{19}$$

where a lower perplexity indicates a higher likelihood that the sequence was the training data.

**Comparing to zlib Compression (Zlib):** This method measures the entropy of a sequence when compressed using zlib, compares the perplexity of a model to its zlib compression entropy, and uses their ratio as an inference metric Carlini et al. (2021).

**Comparing to Lowercased Text (Lowercase):** This method evaluates the change in perplexity of a sequence before and after it has been lowercased, to assess the model's dependency on specific capitalization Carlini et al. (2021):

$$\text{Perplexity Ratio} = \frac{\mathcal{P}(\text{Original})}{\mathcal{P}(\text{Lowercased})}. \tag{20}$$

**Comparing to Other Neural Language Models (Ref):** This approach consists of comparing the ease of error of sequences between the target model and another small model. In our experiments, we specifically use GPT2 as the small model. Note that our approach uses conditional probabilities, whereas Ref does not.

**MIN-K% PROB (MIN-K):** This method Shi et al. (2024) focuses on the minimum token probabilities within a text. It posits that non-member examples are more likely to contain outlier words with high negative log-likelihoods:

$$\text{MIN-K}(x) = \frac{1}{E} \sum_{x_i \in \text{Min-K\%}(x)} \log \pi_\theta(x_i | x_1, ..., x_{i-1}). \tag{21}$$

By analyzing these low probability tokens, MIN-K% PROB provides a distinct method to infer membership, enhancing the diversity of our baseline comparisons.

# B PROOFS

## B.1 Proof of Proposition 1 and 2

Here we give the proof of Proposition 1. Proof of Proposition 2 follows from Proposition 1 of Li et al. (2023). First we give an upper bound on the difference between theoretical maximum expected reward and expected DPO reward on the preference dataset i.e. $r(\pi_r^*) - r(\pi_{\text{DPO}})$

$$\begin{aligned}
& r(\pi_r^*) - r(\pi_{\text{DPO}}) \\
&= \hat{\mathbb{E}}_x \hat{\mathbb{E}}_{y\sim\pi_r^*}[r(x,y)] - \hat{\mathbb{E}}_x \hat{\mathbb{E}}_{y\sim\pi_{\text{DPO}}}[r(x,y)] && (22)\\
&= \hat{\mathbb{E}}_x \hat{\mathbb{E}}_{y\sim\pi_r^*}[r(x,y)] - \hat{\mathbb{E}}_x \hat{\mathbb{E}}_{y\sim\pi_r^*}[\hat{r}_d(x,y)] + \hat{\mathbb{E}}_x \hat{\mathbb{E}}_{y\sim\pi_r^*}[\hat{r}_d(x,y)] \\
&- \hat{\mathbb{E}}_x \hat{\mathbb{E}}_{y\sim\pi_{\text{DPO}}}[r(x,y)] + \hat{\mathbb{E}}_x \hat{\mathbb{E}}_{y\sim\pi_{\text{DPO}}}[\hat{r}_d(x,y)] - \hat{\mathbb{E}}_x \hat{\mathbb{E}}_{y\sim\pi_{\text{DPO}}}[\hat{r}_d(x,y)] && (23)\\
&\leq 2\varepsilon_r + \hat{\mathbb{E}}_x \hat{\mathbb{E}}_{y\sim\pi_r^*}[\hat{r}_d(x,y)] - \hat{\mathbb{E}}_x \hat{\mathbb{E}}_{y\sim\pi_{\text{DPO}}}[\hat{r}_d(x,y)] \qquad \text{(by definition of } \varepsilon_r) && (24)\\
&\leq 2\varepsilon_r \qquad \text{(by definition of } \pi_{DPO}) && (25)
\end{aligned}$$

Similarly we give an upper bound on the difference between theoretical maximum expected reward and expected PPO reward on the preference dataset i.e. $r(\pi_r^*) - r(\pi_{\text{PPO}})$

$$\begin{aligned}
& r(\pi_r^*) - r(\pi_{\text{PPO}}) \\
&= \hat{\mathbb{E}}_x \hat{\mathbb{E}}_{y\sim\pi_r^*}[r(x,y)] - \hat{\mathbb{E}}_x \hat{\mathbb{E}}_{y\sim\pi_{\text{PPO}}}[r(x,y)] && (26)\\
&= \hat{\mathbb{E}}_x \hat{\mathbb{E}}_{y\sim\pi_r^*}[r(x,y)] - \hat{\mathbb{E}}_x \hat{\mathbb{E}}_{y\sim\pi_r^*}[\hat{r}_p(x,y)] + \hat{\mathbb{E}}_x \hat{\mathbb{E}}_{y\sim\pi_r^*}[\hat{r}_p(x,y)] \\
&- \hat{\mathbb{E}}_x \hat{\mathbb{E}}_{y\sim\pi_{\text{PPO}}}[r(x,y)] + \hat{\mathbb{E}}_x \hat{\mathbb{E}}_{y\sim\pi_{\text{PPO}}}[\hat{r}_p(x,y)] - \hat{\mathbb{E}}_x \hat{\mathbb{E}}_{y\sim\pi_{\text{PPO}}}[\hat{r}_p(x,y)] && (27)\\
&\leq 2\varepsilon_r + \hat{\mathbb{E}}_x \hat{\mathbb{E}}_{y\sim\pi_r^*}[\hat{r}_p(x,y)] - \hat{\mathbb{E}}_x \hat{\mathbb{E}}_{y\sim\pi_{\text{PPO}}}[\hat{r}_p(x,y)] \qquad \text{(by definition of } \varepsilon_r) && (28)\\
&\leq 2\varepsilon_r + \hat{\mathbb{E}}_x \hat{\mathbb{E}}_{y\sim\pi_r^*}[\hat{r}_p(x,y)] - \hat{\mathbb{E}}_x \mathbb{E}_{y\sim\pi_r^*}[\hat{r}_p(x,y)] + \hat{\mathbb{E}}_x \mathbb{E}_{y\sim\pi_r^*}[\hat{r}_p(x,y)] \\
&- \hat{\mathbb{E}}_x \hat{\mathbb{E}}_{y\sim\pi_{\text{PPO}}}[\hat{r}_p(x,y)] + \hat{\mathbb{E}}_x \mathbb{E}_{y\sim\pi_{\text{PPO}}}[\hat{r}_p(x,y)] - \hat{\mathbb{E}}_x \mathbb{E}_{y\sim\pi_{\text{PPO}}}[\hat{r}_p(x,y)] && (29)\\
&\leq 2\varepsilon_r + 2\varepsilon_y \qquad \text{(by definition of } \varepsilon_y \text{ and definition of } \pi_{PPO}) && (30)
\end{aligned}$$

## B.2 Proof of Proposition 3

The loss term $\ell(\theta, z_1)$ in equation 13 refers to the negative reward $-\hat{r}(x_1, y)$ in equations 2 and 5. If $\pi_{DPO}$ is indeed the global optimizer, then on the preference dataset $\pi_{DPO}$ is guaranteed to pick the $y \in \{y_w, y_l\}$ such that $-\hat{r}(x_1, y)$ is minimized. If this is not true, then $\pi_{DPO}$ is not the global optimizer. On the other hand, no such condition can be imposed on $\pi_{PPO}$, since the response $y$ is sampled on the conditional distribution of the prompt $x$. Since $\tau$ is assumed to be a constant in equation 13, loss term for $\pi_{DPO}$ will be at least as low as that of $\pi_{PPO}$ for $z_1$. Hence, the result follows as $\sigma$ is monotonic.

### B.3 Proof of Lemma 2

We assume that $n$ iid points are used for training and $\theta$ is the set of parameters obtained after training. We are interested in inferring the membership $m_1$ of $z_1$. We collect the information related to rest of datapoints in $\mathcal{T} = \{z_2, \dots, z_n\}$. From theorem 1 of Sablayrolles et al. (2019), we have that the bayes optimal membership $m_1$ of a sample $z_1$ given $\theta$ is given by

$$\mathcal{M}(\theta, z_1) = \mathbb{P}(m_1 = 1 \mid \theta, z_1) \tag{31}$$
$$= \mathbb{E}_{\mathcal{T}}\left[\mathbb{P}(m_1 = 1 \mid \theta, z_1, \mathcal{T})\right] \tag{32}$$

$$\mathbb{P}(m_1 = 1 \mid \theta, z_1, \mathcal{T}) = \frac{\mathbb{P}(\theta \mid m_1 = 1, z_1, \mathcal{T})\,\mathbb{P}(m_1 = 1)}{\mathbb{P}(\theta \mid z_1, \mathcal{T})} \tag{33}$$
$$= \frac{\alpha}{\alpha + \beta} = \sigma\left(\log\left(\frac{\alpha}{\beta}\right)\right) \tag{34}$$
$$where \qquad \alpha := \mathbb{P}(\theta \mid m_1 = 1, z_1, \mathcal{T})\,\mathbb{P}(m_1 = 1)\,; \beta := \mathbb{P}(\theta \mid m_1 = 0, z_1, \mathcal{T})\,\mathbb{P}(m_1 = 0) \tag{35}$$

If we assume a uniform prior over $m_1$, then we have

$$\log\left(\frac{\alpha}{\beta}\right) = \log\left(\frac{\mathbb{P}(\theta \mid m_1 = 1, z_1, \mathcal{T})}{\mathbb{P}(\theta \mid m_1 = 0, z_1, \mathcal{T})}\right) \tag{36}$$

Next, Sablayrolles et al. (2019) defines a posterior $p(\theta)$ based on the likelihood obtained from $\{z_1, \dots, z_n\}$. However, for conveninence in later steps, we define the posterior as follows:

$$p_{\mathcal{T}}(\theta) := \frac{e^{-\frac{1}{T}\sum_{i=1}^{n} m_i \ell(\theta, z_i)}}{\int_t e^{-\frac{1}{T}\sum_{i=1}^{n} m_i \ell(t, z_i)}\, dt} \tag{37}$$

It is straightforward to see that

$$\alpha = p_{\mathcal{T}}(\theta) \tag{38}$$
$$\beta = \frac{e^{\frac{\ell(\theta, z_1)}{T}} p_{\mathcal{T}}(\theta)}{\int_t e^{\frac{\ell(t, z_1)}{T}} p_{\mathcal{T}}(t) dt} \tag{39}$$

Thus the bayes optimal membership is given by

$$\mathcal{M}(\theta, z_1) = \mathbb{E}_{\mathcal{T}}\left(\sigma\left(\log\left(\int_t e^{\frac{\ell(t, z_1)}{T}} p_{\mathcal{T}}(t) dt\right) - \frac{\ell(\theta, z_1)}{T}\right)\right) \tag{40}$$

### B.4 Proof of Lemma 3

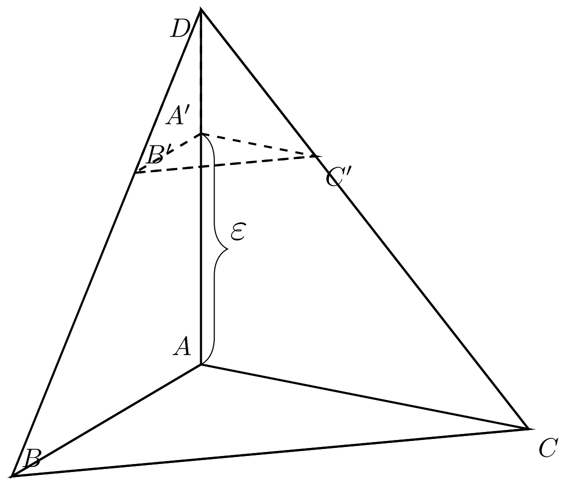



Figure 4: The standard 2-dimensional simplex forms a standard 3-dimensional tetrahedron with the origin $A$ whose volume is given by $\frac{1}{3!}$. The smaller tetrahedron $A'B'C'D$ has a volume $\frac{(1-\varepsilon)^3}{3!}$

Since $L_n \leq \eta$, we have $\sum_{i=1}^{n} \ell(\theta, z_i) < n\eta$, with each individual term in the summation $\geq 0$. Let $\ell_i = \ell(\theta, z_i)$ and since each $z_i$ is an iid sample, the space of $\{\ell_i\}_{i=1}^{n}$ forms a $n$-dimensional standard tetrahedron $\mathcal{V}$ of volume $\frac{(n\eta)^n}{n!}$. Within this tetrahedron $\mathcal{V}$, the space where $\{\ell_1 > \varepsilon\}$ forms a similar tetrahedron of volume maximum $\frac{(n\eta-\varepsilon)^n}{n!}$. For a geometric intuition, refer to Figure 4. Thus, the probability of the event $\{\ell_1 < \varepsilon\}$ is given by

$$\mathbb{P}\left[\{\ell_1 < \varepsilon\}\right] = 1 - \mathbb{P}\left[\{\ell_1 \geq \varepsilon\}\right] \tag{41}$$

$$\geq 1 - \frac{(n\eta - \varepsilon)^n}{(n\eta)^n} \tag{42}$$

$$\geq \left(1 - \left(1 - \frac{1}{n\alpha}\right)^n\right) \qquad \text{(by substituting } \eta = \varepsilon\alpha\text{)} \tag{43}$$

Below we comparison of the lower bounds given here and that of Aubinais et al. (2023).

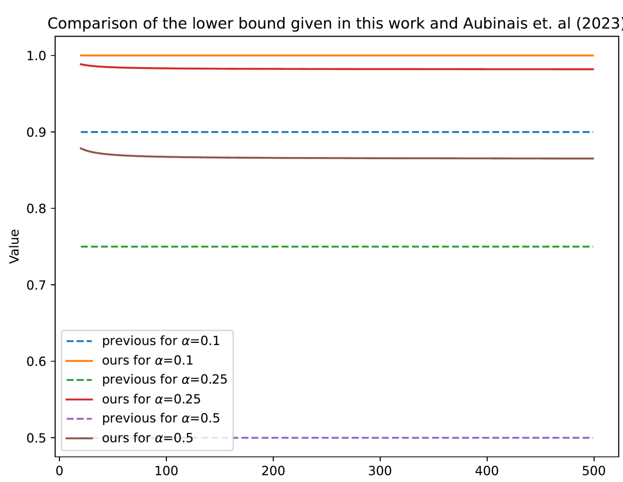



Figure 5: The lower bound given here is much tighter compared to that of Aubinais et al. (2023)

### B.5 Proof of Theorem 2.1

*Proof.* Proof of theorem 2.1

Let $B_\varepsilon := \{\theta : \ell(\theta, z_1) \leq \varepsilon\}$. Then,

$$s(z_1, \theta, p_{\mathcal{T}}) = \log\left(\int_t e^{\frac{1}{T}\ell(t,z_1)} p_{\mathcal{T}}(t)dt\right) - \frac{\ell(\theta, z_1)}{T} \qquad \text{(ignoring } 1/T \text{ for the time being)} \tag{44}$$

$$= \log\left(\int_{B_\varepsilon} e^{\frac{1}{T}\ell(t,z_1)} p_{\mathcal{T}}(t)dt + \int_{B_\varepsilon^C} e^{\frac{1}{T}\ell(t,z_1)} p_{\mathcal{T}}(t)dt\right) - \frac{\ell(\theta, z_1)}{T} \tag{45}$$

$$\geq \log\left(\left(1 - \left(1 - \frac{1}{n\alpha}\right)^n\right) + \int_{B_\varepsilon^C} e^{\frac{1}{T}\ell(t,z_1)} p_{\mathcal{T}}(t)dt\right) - \frac{\ell(\theta, z_1)}{T} \quad \text{(substituting } \ell = 0 \text{ for all } \theta \in B_\varepsilon \text{ and from Lemma 3)} \tag{46}$$

$$\geq \log\left(1 - \left(1 - \frac{1}{n\alpha}\right)^n\right) - \frac{\ell(\theta, z_1)}{T} \quad \text{(dropping the second integral term which} \geq 0 \text{ almost surely)} \tag{47}$$

□

## C IMPLEMENTATION DETAILS

We mainly refer to the TRL* package for implementation.

**LoRA Setting.** For all experiments, we share the same LoRA setting below, using the PEFT* package: `lora_alpha` 32, `lora_dropout` 0.05, `lora_r` 16, and no bias term.

**Quantization Setting.** For all experiments, we use the BitsAndBytes* package for 4-bit quantization.

**SFT Setting.** The settings for SFT are detailed below. We utilized the "train/rl" split of the stack-exchange-paired dataset, selecting 80,000 data points for the fine-tuning process, same data is used for PPO and DPO training. The prompt and only the preferred response are concatenated as input. The specific training parameters are:
- **Training Epochs:** 2.0
- **Learning Rate:** 8e-5
- **Batch Size (Training):** 4
- **Batch Size (Evaluation):** 2
- **Gradient Accumulation Steps:** 4
- **Learning Rate Scheduler:** cosine
- **Warmup Steps:** 100
- **Weight Decay:** 0.05
- **Optimizer:** paged_adamw_32bit
- **Mixed Precision Training:** fp16

**PPO Setting.** The settings for PPO are detailed below. We filter out data points with maximum length constraints. We also limit the maximum length of the generated response. The specific training parameters are:
- **Batch Size:** 16
- **Mini Batch Size:** 4
- **Gradient Accumulation Steps:** 4
- **PPO Epochs:** 6
- **Learning Rate:** 5.4e-5
- **KL Coefficient:** 0.1
- **Adaptive KL Control:** True
- **Target KL:** 5.0

*`https://huggingface.co/docs/trl/en/index`

*`https://huggingface.co/docs/peft/index`

*`https://huggingface.co/docs/bitsandbytes/index`

- **Horizon:** 4000
- **Training Epochs:** 4
- **Maximum Output Length:** 128
- **Maximum Prompt Length:** 256
- **Maximum Sequence Length:** 1024

**DPO Setting.** The settings for DPO training are detailed below. The specific training parameters are:
- **Batch Size (Training):** 8
- **Batch Size (Evaluation):** 2
- **Gradient Accumulation Steps:** 2
- **Training Epochs:** 3.0
- **Learning Rate:** 5e-4
- **Warmup Steps:** 100
- **Maximum Sequence Length:** 1024
- **Maximum Prompt Length:** 256
- **Optimizer Type:** paged_adamw_32bit
- **Beta:** 0.4

# D ADDITIONAL EXPERIMENTAL RESULTS

## D.1 Results of Experiments on GPT2 Series

In Table 5, we give the AUROC scores for other frameworks with GPT2 series on SE dataset. Compared to the optimistic PREMIA, we find that the traditional MIA methods are not that effective when it comes to MIA on smaller models such as GPT2-series. Nevertheless, here too, we consistently notice that susceptibility of DPO is higher compared to PPO.

Table 5: AUROC scores comparing different MIA methods on GPT2-series on SE dataset

| | | Chosen | | Rejected | |
|---|---|---|---|---|---|
| | Framework | DPO | PPO | DPO | PPO |
| Modelname | | | | | |
| gpt2 | MIN-K | 0.545 | 0.526 | 0.523 | 0.524 |
| | PPL | 0.538 | 0.528 | 0.519 | 0.524 |
| | Ref | 0.556 | 0.521 | 0.542 | 0.521 |
| | Lowercase | 0.559 | 0.519 | 0.523 | 0.532 |
| | Zlib | 0.520 | 0.518 | 0.528 | 0.532 |
| gpt2-medium | MIN-K | 0.516 | 0.515 | 0.522 | 0.530 |
| | PPL | 0.512 | 0.516 | 0.517 | 0.529 |
| | Ref | 0.526 | 0.516 | 0.543 | 0.508 |
| | Lowercase | 0.543 | 0.514 | 0.528 | 0.521 |
| | Zlib | 0.528 | 0.525 | 0.514 | 0.520 |
| gpt2-large | MIN-K | 0.525 | 0.517 | 0.537 | 0.518 |
| | PPL | 0.521 | 0.518 | 0.531 | 0.520 |
| | Ref | 0.528 | 0.527 | 0.537 | 0.528 |
| | Lowercase | 0.566 | 0.529 | 0.538 | 0.528 |
| | Zlib | 0.519 | 0.527 | 0.524 | 0.512 |
| gpt2-xl | MIN-K | 0.524 | 0.521 | 0.536 | 0.514 |
| | PPL | 0.519 | 0.518 | 0.533 | 0.509 |
| | Ref | 0.528 | 0.511 | 0.549 | 0.518 |
| | Lowercase | 0.566 | 0.506 | 0.538 | 0.512 |
| | Zlib | 0.520 | 0.525 | 0.520 | 0.528 |

## D.2 Impact of Response Length on MIA Effectiveness

In this experiment, we look the effect of length of examples used in preference alignment and their corresponding vulnerability in terms of AUC-ROC of PREMIA-SFT. Figure 6 shows the MIA AUROC results for the GPT-2 family of models on the IMDB dataset. As can be seen from the figure, for "Chosen" responses, the longer the response, the more susceptible it is to MIA, while for "Rejected" responses, the opposite is true. Note that in general, increasing sequence length leads to higher AUC as longer texts tend to have more memorized information by the model (Shi et al., 2024). However, the 'Chosen' response is used during both SFT and DPO alignment whereas the 'Rejected' response is used only during DPO, thus the difference in their trends when measured using PREMIA-SFT. Since in our PREMIA-SFT framework, we divide by the conditional probability of the SFT model, the higher probability of the longer texts in DPO gets negated by the higher probability in SFT model.

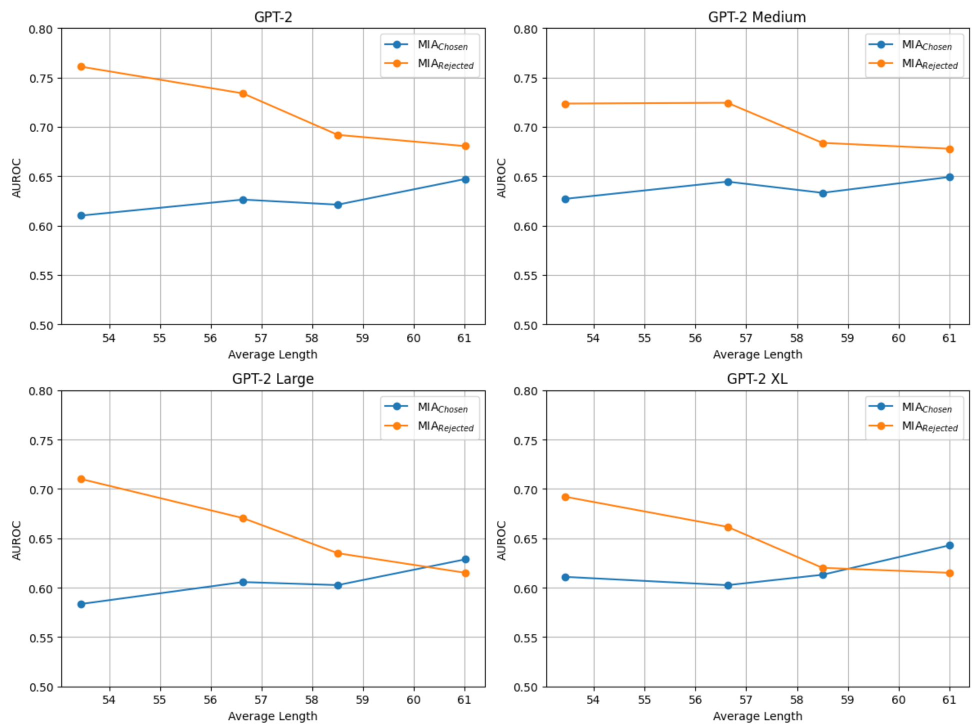
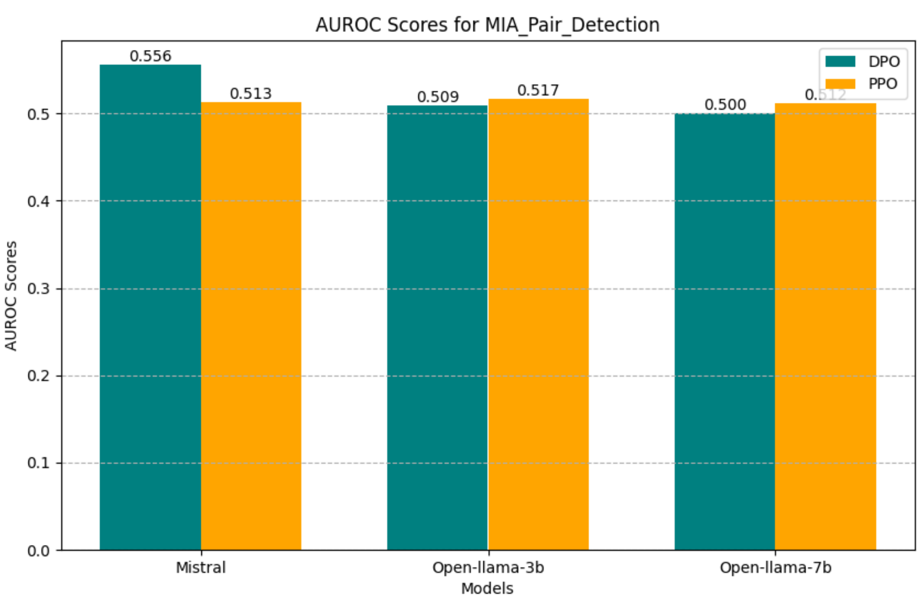



Figure 6: AUROC vs Average Length for GPT-2 Models

### D.3 AUROC plot for pair detection for IMDB dataset

Figure 7: AUROC scores for $MIA_{Pair}$ detection for Mistral-7B, Open-llama-3b, and Open-llama-7b models for IMDB dataset. As mentioned in §4.3.1, for easier tasks such as the IMDB sentiment alignment, both PPO and DPO are robust to MIA.



### D.4 Anecdotal responses by Mistral-7B models on SE dataset

| | |
|---|---|
| Question | What are some grilling tools that a beginner should have when purchasing a new grill? Additionally, are there tools that are useful for the more advanced outdoor cook? |
| Chosen | Tool-wise: spatula, tongs, grill brush (for cleaning) and a basting brush. If you're using extremely hot fire or high flames there are grill-safe mitts you can buy. |
| Rejected | I have a wide, flat knife that can be used as a spatula; I find it really useful. Everything else has pretty much been mentioned, but I would suggest a pair of tongs with decent teeth on them. It's very disheartening when that burger or steak slips out of your grip and into the coals or onto the ground. |
| Base | 1. A good set of tongs.<br>2. A good set of spatulas.<br>3. A good set of skewers.<br>4. A good set of grill brushes.<br>5. A good set of grill gloves.<br>6. A good set of grill tools.<br>7. A good set of grill accessories.<br>8. A good set of grill accessories.<br>9. A good set of grill accessories. |
| SFT | 1. A good set of tongs.<br>2. A good set of spatulas.<br>3. A good set of skewers.<br>4. A good set of grill brushes.<br>5. A good set of grill gloves.<br>6. A good set of grill tools.<br>7. A good set of grill baskets.<br>8. A good set of grill baskets.<br>9. A good set of grill baskets.<br>10. A good set of grill baskets.<br>11. A good set of grill baskets. |
| DPO | **Beginner:**<br>* **Tongs:** These are the most important tool. You'll use them for everything from flipping burgers to moving hot coals. Get a pair that are comfortable to hold and have a good grip. * **Spatula:** This is for flipping things like pancakes and eggs. You'll also use it to scrape the grill grates clean. * **Grill brush:** This is for cleaning the grill grates. You'll want to get one with stiff bristles. * **Thermometer:** This is for checking the temperature of the grill. You'll want to get a digital one that has a probe.<br>**Advanced:**<br>* **Skewers:** These are for making kebabs. * **Basting brush:** This is for brushing sauces on meat. * **Grill basket:** This is for cooking vegetables and other things that are small and fall through the grates. * **Grill mat:** This is for cooking things that are delicate and fall through the grates. * **Grill gloves:** These are for protecting your hands from the heat. * **Grill light:** This is for seeing what you're doing at night.<br>**Both:**<br>* **Grill cleaner:** This is for cleaning the grill grates. * **Grill cover:** This is for protecting the grill from the elements. |
| PPO | 1. A good set of tongs<br>2. A good set of grill brushes<br>3. A good set of grill tools (spatula, fork, etc.)<br>4. A good set of grill gloves<br>5. A good set of grill mats<br>6. A good set of grill baskets |

| | |
|---|---|
| Question | Does ‘dbms_output.put_line‘ decrease the performance in ‘plsql‘ code? |
| Chosen | Yes, it’s another piece of code that needs to be executed, but unless the output is actually turned on, I think the overhead is quite minimal. Here’s an AskTom question with more details: Is there a performance impact for dbms_output.put_line statements left in packages? `http://asktom.oracle.com` |
| Rejected | You can look into [conditional compilation](http://www.oracle-base.com/articles/10g/ConditionalCompilation_10gR2.php) so that the DBMS_OUTPUT.PUT_LINE are only in the pre-parsed code if the procedure is compiled with the appropriate option. |
| Base | No, it does not.<br>‘dbms_output‘ is a package that is used to display the output of a PL/SQL block. It is not a performance issue.<br>‘dbms_output‘ is a package that is used to display the output of a PL/SQL block. It is not a performance issue.<br>The ‘dbms_output‘ package is used to display the output of a PL/SQL block. It is not a performance issue.<br>The ‘dbms_output‘ package is used to display the output of a PL/SQL block. It is not a performance issue.<br>The ‘dbms_output‘ package is used to display the output of a PL/SQL block. It is not a performance issue. |
| SFT | 1. ‘dbms_output‘ is a package that is used to display output from PL/SQL blocks. 2. ‘dbms_outputput_line‘ is a procedure that is used to display a line of output from PL/SQL blocks. 3. ‘dbms_output‘ is not a function, it is a package. 4. ‘dbms_output.put_line‘ is not a function, it is a procedure. 5. ‘dbms_output‘ is not a variable, it is a package. 6. ‘dbms_output.put_line‘ is not a variable, it is a procedure. 7. ‘dbms_output‘ is not a constant, it is a package. 8. ‘dbms_output.put_line‘ is not a constant, it is a procedure. |
| DPO | **Yes, it does.**<br>‘dbms_output‘ is meant for debugging only. It is not meant for production code.<br>From the documentation: > **dbms_output** The DBMS_OUTPUT package provides a mechanism for |
| PPO | Yes, it does.<br>‘dbms_output‘ is a package that is used to display output from a PL/SQL block. It is not a package that is used to display output from a SQL statement.<br>The ‘dbms_output‘ package is a package that is used to display output from a PL/SQL block. |