Private Data Synthesis for Preference Learning in Large Language Models

Fengyu Gao **Wei Shen**

Abstract

Preference learning has become a crucial technique for aligning large language models (LLMs) with human values. However, training on real human preference data raises privacy concerns, as these datasets often contain sensitive user prompts and human judgments. To address this, we propose a **two-phase framework** for privacy-preserving preference learning. In **Phase 1**, we introduce the **DPPref-Syn** algorithm that generates differentially private (DP) synthetic preference data. DPPrefSyn addresses three key challenges: modeling diverse human preferences via clustering and per-cluster DP scoring models; reducing dimensionality with DP-PCA to improve efficiency; and conserving privacy budget by leveraging public prompts. In **Phase 2**, we fine-tune an LLM using preference learning methods on the synthetic data produced in Phase 1. We conduct extensive experiments on standard benchmarks and compare our method with non-private preference learning and base LLMs. Results show that it achieves competitive performance under strong privacy guarantees. For example, on Anthropic-HH with DPO, our method achieves a GPT-4 win rate of 51.72% using only 5K public prompts and $\varepsilon = 4$, outperforming direct training on private data (38.00%). These results open up new possibilities for preference learning with privacy protection for a broad range of applications. To the best of our knowledge, this is the first work to generate DP synthetic preference data for LLM alignment.

1 Introduction

Preference learning algorithms—such as RLHF (Stiennon et al., 2020) and DPO (Rafailov et al., 2024)—are widely used to align Large Language Models (LLMs) with human expectations. This is typically achieved by collecting a dataset of pairwise preferences, where human annotators indicate which of two responses better answers a given prompt. Such preference data is then used to fine-tune LLMs by encouraging the model to rank preferred responses over dispreferred ones (Ziegler et al., 2019; Ouyang et al., 2022; Bai et al., 2022). The effectiveness and reliability of preference learning have motivated its adoption in applications such as chat assistants (Achiam et al., 2023), mathematical reasoning tools (Shao et al., 2024), and code generators (Shen et al., 2023).

However, privacy is a significant concern when aligning LLMs with human preference data, as it contains real user prompts and feedback. These prompts may disclose personal information related to health, identity, or other sensitive topics, and human feedback can reveal private beliefs, preferences, or behavioral patterns (Li et al., 2023; Yu et al., 2024). To address this concern, existing works (Chowdhury et al., 2024; Yu et al., 2024; Wu et al., 2023a) have explored approaches to mitigate these risks using the rigorous privacy safeguards provided by differential privacy (DP) (Dwork et al., 2006b) and shown encouraging results. However, many of these approaches protect only part of the data—for example, by privatizing either the user prompts (Yu et al., 2024) or the preference labels (Chowdhury et al., 2024; Zhang et al., 2025), but not both. One line of work focuses on specific training algorithms, such as PPO (Wu et al., 2023a), and is not compatible with newer methods like DPO. These gaps lead to the central question of this work:

Preprint.

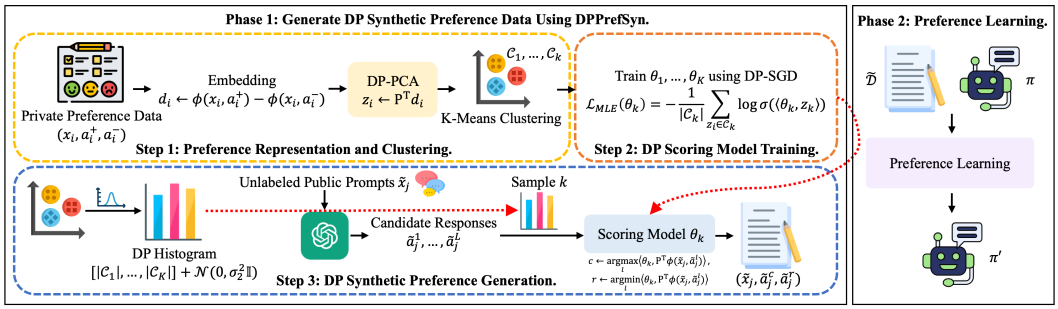

Figure 1: Our two-phase framework for privacy-preserving preference learning. Phase 1 generates DP synthetic preference data using DPPrefSyn. This phase involves (1) representing preference samples as embedding differences and clustering them via DP-PCA and K-means, (2) training DP scoring models on each cluster using DP-SGD, and (3) generating synthetic preference samples from public prompts guided by the DP scoring models and DP histogram. Phase 2 fine-tunes a downstream model using the synthetic preference dataset.

Can we fulfill the promise of preference learning while ensuring thorough privacy protection and remaining compatible with evolving preference learning methods?

In this work, we provide an affirmative answer to this question. Our main contributions are summarized as follows:

- We propose a new two-phase framework for privacy-preserving preference learning (Figure 1). In **Phase 1**, we introduce DPPrefSyn (Algorithm 1), an algorithm for generating DP synthetic preference data. DPPrefSyn features the following elements in its design: 1) *Modeling preference heterogeneity via clustering.* We represent each preference sample using the embedding difference between concatenated prompt-preferred and prompt-dispreferred pairs, and apply clustering to group samples with similar preference patterns. This enables learning per-cluster scoring models that better reflect diverse human preferences. 2) *Improving efficiency through DP-PCA.* To reduce the cost of training multiple models on high-dimensional embeddings, we apply DP-PCA to project the data into a lower-dimensional space while preserving key preference signals under DP. 3) *Saving privacy budgets by using public prompts.* We use public prompts to generate candidate responses and apply private scoring models to select preference pairs. This allows us to focus the privacy budget entirely on modeling private preferences, improving privacy efficiency. In **Phase 2**, we fine-tune an LLM using preference learning algorithms (e.g., DPO) on the synthetic preference data generated in Phase 1.

- We conduct a rigorous privacy analysis of our framework and confirm it follows (ε, δ)-DP. All privacy costs are incurred in Phase 1, where we first allocate a privacy budget of ε_0-DP to DP-PCA (Amin et al., 2019) for dimensionality reduction. The remaining budget is used to train per-cluster scoring models with DP-SGD and privatize the cluster histogram used for sampling. We apply the Privacy Random Variable (PRV) accountant (Gopi et al., 2021) to tightly compose the privacy guarantees of these mechanisms. Thanks to the post-processing property of DP, Phase 2, which fine-tunes an LLM using the synthetic dataset, incurs no additional privacy cost. Therefore, the DP synthetic dataset can be reused across multiple preference learning methods and different LLMs.

- We empirically evaluate our DP framework on standard benchmarks, including question answering tasks from OpenAssistant (Köpf et al., 2023) and Anthropic-HH (Bai et al., 2022), as well as the TL;DR summarization task (Stiennon et al., 2020). Our experiments suggest that our framework supports privacy-preserving preference learning, offering competitive utility while ensuring strong privacy protections. As a representative example, using only 5K public prompts from SafeRLHF (Ji et al., 2024) and a privacy budget of $\varepsilon = 4$, our method achieves a GPT-4 win rate of 51.72% on Anthropic-HH, outperforming a strong baseline that directly fine-tunes on private data using DPO (38.00%). These results show that synthetic preference data generated under DP can enable both effective and private preference learning for LLMs.

2 Related Work

Differentially Private Alignment of LLMs. In the context of privacy-preserving preference learning for LLM alignment, Wu et al. (2023a) first introduce a DP framework to align LLMs with reinforcement learning by adapting the PPO algorithm to the DP setting. However, their approach is *limited to PPO*, while preference learning continues to evolve with new methods. Chowdhury et al. (2024) study the problem of reward estimation from preference-based feedback, using the notion of label-DP (Ghazi et al., 2021) to protect the privacy of human annotators. Zhang et al. (2025) propose AUP-RLHF, a user-level label-DP framework for RLHF. While both methods are effective at safeguarding preference labels, they *do not address the sensitivity of prompts or responses themselves*. Yu et al. (2024) focus on the privacy risks of using sensitive user instructions in LLM alignment and propose to generate differentially private synthetic *instructions* to replace real ones during data annotation and model fine-tuning.

Beyond preference learning, several studies focus on private fine-tuning of LLMs (Yu et al., 2021; Li et al., 2021; Chen et al., 2024; Tang et al., 2024; Zhang et al., 2023; He et al., 2022), and some recent work focuses on protecting the privacy of in-context prompts for in-context learning (Duan et al., 2023; Tang et al., 2023; Gao et al., 2024; Hong et al., 2023; Wu et al., 2023b).

Differentially Private Synthetic Text Generation. Our work falls within the broader scope of DP synthetic text generation. In this area, two representative lines of work have emerged. The first line fine-tunes a language model on private data under DP, and then uses the fine-tuned model to generate synthetic text (Yue et al., 2023; Mattern et al., 2022; Mireshghallah et al., 2023; Carranza et al., 2023; Yu et al., 2024; Wang et al., 2024; Ochs and Habernal, 2024; Tan et al., 2025; Carranza et al., 2024). For example, Yu et al. (2024) apply LoRA fine-tuning to LLaMA-7B and LLaMA-13B models on private instruction data under DP for supervised instruction tuning.

Since DP fine-tuning can be expensive and sometimes infeasible (especially for non-public language models), a recent line of work explores DP synthetic data generation by prompting LLM APIs (Lin et al., 2023; Xie et al., 2024; Wu et al., 2024; Hou et al., 2024). These approaches typically generate synthetic samples using LLM API access and then select outputs that are similar to the private data in a privacy-preserving manner. For instance, Xie et al. (2024) propose the Aug-PE algorithm, which begins with generating initial synthetic samples using LLM APIs and human-crafted prompts, then iteratively refines them by selecting examples close to private data in embedding space using a DP nearest neighbor histogram. In this work, we follow the general approach in the second line of work but focus on the preference learning domain. We train linear models on clustered private preference data under DP, and use these models to score and select preferred and less preferred responses from LLM-generated candidates. To the best of our knowledge, this is the first work to generate a DP synthetic preference dataset for preference learning.

Diversity in Human Preferences. Recent works highlight that human preferences are inherently diverse (Denton et al., 2021; Aroyo et al., 2023a,b; Chakraborty et al., 2024). The key factors include sociodemographic differences (e.g., race, gender, age), personal beliefs and biases, varying levels of domain expertise, and the inherent ambiguity of natural language (Sandri et al., 2023; Vogels, 2021). Motivated by this, recent studies have explored personalizing preference learning to better reflect the values of different user groups (Lee et al., 2024; Poddar et al., 2024; Singh et al., 2025). Lee et al. (2024) train an ensemble (a single neural network with multiple prediction heads), and dynamically reweight them at test time using a small set of labeled examples from the target distribution. Poddar et al. (2024) model user-specific latent variables and learn reward models and policies conditioned on the latents. Singh et al. (2025) propose a meta-learning framework where an LLM adapts to individual users using a small number of preference examples. In this work, we generate DP synthetic preference data that capture the underlying diversity of human feedback, enabling their use in different preference learning methods, including those aimed at personalization.

3 Problem Definition, Threat Model, and Notations

Preference learning in LLMs. We denote an LLM by a policy π that maps a prompt $x \in \mathcal{X}$ to a distribution over possible responses in $\mathcal{A}$. In a typical preference learning pipeline, a pretrained LLM π_0 is first fine-tuned using supervised learning (SFT) to adapt it to a downstream task or desired output style, resulting in π_{SFT}. To further align π_{SFT} with human preferences, a preference dataset

$\mathcal{D} = \{(x_i, a_i^+, a_i^-)\}_{i=1}^n$ is collected. Each sample consists of a prompt $x_i \in \mathcal{X}$, and two responses $a_i^+, a_i^- \in \mathcal{A}$ generated by a language model, where a_i^+ is labeled as the preferred response and a_i^- as the less preferred one by human annotators. This preference data is then used to fine-tune π_{SFT} via methods such as reward modeling and reinforcement learning (e.g., RLHF), or direct optimization approaches like DPO. Following Ouyang et al. (2022); Zhu et al. (2023); Rafailov et al. (2024); Liu et al. (2023), we assume that there exists a ground truth reward function $r^\star(x, a) : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ and the human preference satisfies the Bradley-Terry model (Bradley and Terry, 1952):

$$\mathbb{P}[a_i^+ \succ a_i^- \mid x_i] = \frac{\exp(r^\star(x_i, a_i^+))}{\exp(r^\star(x_i, a_i^+)) + \exp(r^\star(x_i, a_i^-))} = \sigma(r^\star(x_i, a_i^+) - r^\star(x_i, a_i^-)),$$

where $a_i^+ \succ a_i^-$ means a_i^+ is is preferred to a_i^-, and $\sigma(z) = 1/(1 + e^{-z})$ is the sigmoid function.

In this work, we consider a linear reward model $r_\theta(x, a) = \langle \theta, \phi(x, a) \rangle$, where $\phi : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^d$ is a known mapping and $\theta \in \mathbb{R}^d$ is a learnable parameter (Saha et al., 2023; Kong and Yang, 2022; Zhu et al., 2023). With this model, we can equivalently write:

$$\mathbb{P}[a_i^+ \succ a_i^- \mid x_i] = \sigma(\langle \theta, \phi(x_i, a_i^+) - \phi(x_i, a_i^-) \rangle).$$

The most common algorithm for training such reward models in RLHF is maximum likelihood estimation (MLE) (Christiano et al., 2017; Ouyang et al., 2022; Zhu et al., 2023). Given the preference dataset $\mathcal{D}$, MLE aims at minimizing the negative log likelihood:

$$\mathcal{L}_{\text{MLE}}(\theta) = -\frac{1}{n} \sum_{i=1}^{n} \log \sigma \left(\langle \theta, \phi(x_i, a_i^+) - \phi(x_i, a_i^-) \rangle \right).$$

Differentially private preference learning. Our objective is to protect the privacy of the preference dataset $\mathcal{D}_{\text{priv}} = \{(x_i, a_i^+, a_i^-)\}_{i=1}^n$ against an adversary who attempts to access or infer private information about individual prompts, responses, or preference labels. To achieve this, we propose to generate a DP synthetic preference dataset $\tilde{\mathcal{D}} = \{(\tilde{x}_j, \tilde{a}_j^+, \tilde{a}_j^-)\}_{j=1}^m$ from the underlying distribution of $\mathcal{D}_{\text{priv}}$. This synthetic dataset can then be used to fine-tune LLMs using preference learning methods such as DPO or RLHF. Thanks to the post-processing property of DP, $\tilde{\mathcal{D}}$ can be reused across various preference learning methods and LLMs without incurring additional privacy cost.

We formally define (ε, δ)-differential privacy ((ε, δ)-DP) as follows.

Definition 1 ((ε, δ)-Differential Privacy (Dwork et al., 2006a))**.** A randomized algorithm $\mathcal{A}$ is (ε, δ)-differentially private if for any two neighboring inputs $\mathcal{D}$ and $\mathcal{D}'$ that differ by a single entry and any set $\mathcal{S}$ of possible outputs: $\mathbb{P}[\mathcal{A}(\mathcal{D}) \in \mathcal{S}] \leq e^\varepsilon \mathbb{P}[\mathcal{A}(\mathcal{D}') \in \mathcal{S}] + \delta$.

4 Proposed Method

Our framework for preference learning with DP guarantees consists of two phases:

Phase 1: Generate a DP synthetic dataset $\tilde{\mathcal{D}} = \{(\tilde{x}_j, \tilde{a}_j^+, \tilde{a}_j^-)\}_{j=1}^m$ from the private dataset $\mathcal{D}_{\text{priv}}$;

Phase 2: Use this DP synthetic dataset to fine-tune an LLM using preference learning algorithms.

The post-processing property of DP ensures that the second phase incurs no additional privacy cost. In the following, we focus on the algorithm that implements the first phase.

4.1 Intuition behind Algorithm Design

To motivate our algorithm design, we first discuss the key challenges in synthesizing DP preference data. First, human preferences in the private dataset are not uniform (Chakraborty et al., 2024; Bakker et al., 2022; Kovač et al., 2023; Jang et al., 2023; Rame et al., 2023; Ji et al., 2023). Different annotators may prioritize different aspects of a response, such as factual accuracy, politeness, creativity, or clarity, resulting in diverse and sometimes conflicting signals in the data. Ignoring this diversity in the underlying preferences can result in synthetic data that fail to reflect the true distribution of $\mathcal{D}_{\text{priv}}$. To address this, we first represent each sample in the private preference dataset using the embedding difference between the chosen and rejected responses, conditioned on the same prompt,

i.e., $\phi(x_i, a_i^+) - \phi(x_i, a_i^-)$, where ϕ is an embedding model. These vectors capture the implicit direction of preference expressed in each comparison. We then cluster them into multiple groups with similar preference patterns. Within each cluster, we train a private scoring model to guide the selection of synthetic preference pairs that reflect the cluster-specific structure of human preferences.

Second, the embeddings of (prompt, response) pairs $\phi(x, a)$ are intentionally high-dimensional to capture rich semantic and stylistic features. Since we train a separate scoring model for each cluster of samples with similar preference patterns, learning multiple models on high-dimensional inputs becomes computationally expensive and sample inefficient. To mitigate this, we apply differentially private PCA to reduce the dimensionality of the representations before clustering and training. This improves efficiency and stability, while preserving the core preference signals.

Third, synthesizing prompts, preferred responses and less preferred responses simultaneously can quickly exhaust the privacy budget and degrade utility. To address this, we choose to use public prompts to avoid spending privacy budget on synthesizing DP prompts, allowing us to allocate the entire budget to modeling preferences over responses. For each public prompt, we generate multiple candidate responses using an LLM and apply privately trained scoring functions—learned from private preference data—to construct preference pairs. While public prompts may differ in distribution from private ones, our experiments show that, despite this distribution shift, the synthetic preference data remains effective for downstream preference learning and alignment tasks.

4.2 DPPrefSyn Algorithm

We now introduce the proposed DPPrefSyn algorithm (Algorithm 1), which generates DP synthetic preference dataset from the private dataset $\mathcal{D}_{\text{priv}}$. DPPrefSyn consists of 3 main steps:

Step 1: Preference Representation and Clustering. We encode each prompt-response pair in $\mathcal{D}_{\text{priv}}$ using a sentence embedding model ϕ. Specifically, for each preference sample (x_i, a_i^+, a_i^-), we first concatenate the prompt with each response to form two texts: $[x_i; a_i^+]$ and $[x_i; a_i^-]$. We then compute their embeddings via ϕ, and define the embedding difference as $d_i \leftarrow \phi(x_i, a_i^+) - \phi(x_i, a_i^-)$ (Line 3).

To reduce dimensionality while preserving privacy, we apply DP-PCA (Amin et al., 2019; Shoemate et al., 2021) to $\{d_i\}_{i=1}^n$, obtaining a projection matrix $\mathbf{P} \in \mathbb{R}^{d \times p}$ under ε_0-DP (Line 5). We privatize $\mathbf{P}$ here because it is later used to project public samples; using a projection matrix derived from private data without privatization would violate DP. Roughly speaking, DP-PCA (Amin et al., 2019) approximates the eigendecomposition of the data covariance matrix by estimating the collections of eigenvalues and eigenvectors separately in a DP manner.

Each d_i is then projected to a lower-dimensional space as $z_i \leftarrow \mathbf{P}^\top d_i$ (Line 6). We apply K-means clustering on $\{z_i\}_{i=1}^n$ to group samples with similar preference patterns, forming K clusters $\mathcal{C}_1, \ldots, \mathcal{C}_K$ in Line 7.

Step 2: DP Scoring Model Training. For each cluster $\mathcal{C}_k$, we train a linear scoring model $\theta_k \in \mathbb{R}^p$ that captures preference patterns (Line 9). Specifically, we optimize θ_k by minimizing the negative log-likelihood:

$$\mathcal{L}_{\text{MLE}}(\theta_k) = -\frac{1}{|\mathcal{C}_k|} \sum_{z_i \in \mathcal{C}_k} \log \sigma\left(\langle \theta_k, z_i \rangle\right), \tag{1}$$

where z_i is the DP-PCA projected embedding difference from Step 1. Each scoring model is trained with the DP-SGD algorithm (Abadi et al., 2016) using noise multiplier σ_1. DP-SGD is the most common approach for private deep learning; it ensures DP by applying per-sample gradient clipping and adding carefully calibrated Gaussian noise to the gradient updates during each iteration of training.

Step 3: DP Synthetic Preference Generation. We compute a histogram $\mathbf{h} = [|\mathcal{C}_1|, \ldots, |\mathcal{C}_K|]$ representing the number of samples in each cluster (Line 11). This histogram is made private by adding appropriate Gaussian noise to each bin $\tilde{\mathbf{h}} \leftarrow \mathbf{h} + \mathcal{N}(0, \sigma_2^2 \mathbb{I}_{K \times K})$ (Line 12), and we normalize it to get a probability distribution $\tilde{\mathbf{p}}$ over the clusters, i.e., $\tilde{\mathbf{p}} \leftarrow \tilde{\mathbf{h}}/|\mathcal{D}_{\text{priv}}|$ (Line 13).

Next, for each public prompt $\tilde{x}_j$, we use a relatively high temperature to prompt an LLM to generate L diverse candidate responses $\tilde{a}_j^1, \ldots, \tilde{a}_j^L$ (Line 15). We then sample a cluster index $k \sim \tilde{\mathbf{p}}$ according

to the DP histogram (Line 16) and evaluate the responses $\tilde{a}_j^1, \ldots, \tilde{a}_j^L$ using the scoring model θ_k associated with the sampled cluster k. To do this, we first compute the embeddings $\phi(\tilde{x}_j, \tilde{a}_j^l)$ for $l = 1, \ldots, L$, and project them into the DP-PCA subspace using $\mathbf{P}$. The preference score for $\tilde{a}_j^l$ is then computed as $\langle \theta_k, \mathbf{P}^\top \phi(\tilde{x}_j, \tilde{a}_j^l) \rangle$. We select the highest- and lowest-scoring responses as the preferred and less preferred responses, respectively: $c \leftarrow \arg\max_{l \in [L]} \langle \theta_k, \mathbf{P}^\top \phi(\tilde{x}_j, \tilde{a}_j^l) \rangle$, $r \leftarrow \arg\min_{l \in [L]} \langle \theta_k, \mathbf{P}^\top \phi(\tilde{x}_j, \tilde{a}_j^l) \rangle$ (Line 17). If the score gap is too small (e.g., < 0.5), we discard the sample to ensure preference quality. Otherwise, we add $(\tilde{x}_j, \tilde{a}_j^c, \tilde{a}_j^r)$ to $\tilde{\mathcal{D}}$. This process is repeated for all public prompts, producing a DP synthetic preference dataset $\tilde{\mathcal{D}}$.

Algorithm 1 DPPrefSyn

1: **Input:** Private dataset $\mathcal{D}_{\text{priv}} = \{(x_i, a_i^+, a_i^-)\}_{i=1}^n$, public prompts $\{\tilde{x}_j\}_{j=1}^m$, embedding model ϕ, number of clusters K, an LLM $\text{LLM}(\cdot)$, DP parameters ε_0, σ_1, σ_2.
2: **for** each (x_i, a_i^+, a_i^-) in $\mathcal{D}_{\text{priv}}$ **do**
3: $\quad d_i \leftarrow \phi(x_i, a_i^+) - \phi(x_i, a_i^-)$, where $d_i \in \mathbb{R}^d$
4: **end for**
5: $\mathbf{P} \leftarrow \text{DP-PCA}(\{d_i\}_{i=1}^n, \varepsilon_0)$, where $\mathbf{P} \in \mathbb{R}^{d \times p}$
6: $\{z_i\}_{i=1}^n \leftarrow \{\mathbf{P}^\top d_i\}_{i=1}^n$
7: $\mathcal{C}_1, ..., \mathcal{C}_K \leftarrow \text{K-means}(\{z_i\}_{i=1}^n, K)$
8: **for** each cluster $\mathcal{C}_k$ **do**
9: $\quad$ Train a linear scoring model θ_k on $\mathcal{C}_k$ using DP-SGD with noise multiplier σ_1 (Equation (1))
10: **end for**
11: $\mathbf{h} \leftarrow [|\mathcal{C}_1|, \ldots, |\mathcal{C}_K|]$
12: $\tilde{\mathbf{h}} \leftarrow \mathbf{h} + \mathcal{N}(0, \sigma_2^2 \mathbb{I}_{K \times K})$
13: $\tilde{\mathbf{p}} \leftarrow \tilde{\mathbf{h}} / |\mathcal{D}_{\text{priv}}|$
14: **for** each public prompt $\tilde{x}_j$ **do**
15: $\quad$ Generate candidate responses $\tilde{a}_j^1, \ldots, \tilde{a}_j^L \leftarrow \text{LLM}(\tilde{x}_j)$
16: $\quad$ Sample cluster index $k \sim \tilde{\mathbf{p}}$
17: $\quad c \leftarrow \arg\max_{l \in [L]} \langle \theta_k, \mathbf{P}^\top \phi(\tilde{x}_j, \tilde{a}_j^l) \rangle$, $r \leftarrow \arg\min_{l \in [L]} \langle \theta_k, \mathbf{P}^\top \phi(\tilde{x}_j, \tilde{a}_j^l) \rangle$
18: $\quad$ Add $(\tilde{x}_j, \tilde{a}_j^c, \tilde{a}_j^r)$ to $\tilde{\mathcal{D}}$
19: **end for**
20: **return** $\tilde{\mathcal{D}}$.

4.3 Privacy Analysis

Theorem 1. *Algorithm 1 is (ε, δ)-differentially private.*

Proof Overview. We first allocate a privacy budget of ε_0-DP to DP-PCA (Amin et al., 2019) to generate a projection matrix $\mathbf{P}$ in Line 5. The remaining budget is allocated to 2 components: (1) training per-cluster scoring models using DP-SGD with noise multiplier σ_1 (Line 9), and (2) privatizing the cluster histogram with Gaussian noise $\mathcal{N}(0, \sigma_2^2 \mathbb{I})$ (Line 12). For training DP scoring models, we apply parallel composition of DP: since the clusters are disjoint, modifying a single data point in the training data affects only one cluster. As a result, the total privacy cost is bounded by the cost of training on the smallest cluster used. We ensure a known lower bound on cluster size by discarding clusters with too few samples. The DP histogram step follows from the Gaussian DP mechanism. We use the Privacy Random Variable (PRV) accountant (Gopi et al., 2021) to compose these two steps and ensure that their composition satisfies $(\varepsilon - \varepsilon_0, \delta)$-DP. All other steps involve only public data or post-processing, and incur no additional privacy cost.

In our experiments, we set the total privacy budget to $\varepsilon = 4, 8$, and fix the DP-PCA budget to $\varepsilon_0 = 0.5, 1$, respectively. The default noise multiplier σ_2 for releasing the histogram is 20.0. We then choose noise multiplier σ_1 such that the total privacy loss satisfies (ε, δ)-DP. We provide a full proof and the codes for privacy accounting in Appendix A.

□

5 Experiments

Datasets. We study two types of tasks: question answering and summarization. For question answering, we use the OpenAssistant dataset (Köpf et al., 2023), which contains assistant-style conversations (14K English training examples and 712 test examples), and the Anthropic-HH dataset (Bai et al., 2022), which provides human preference comparisons focused on helpfulness and harmlessness (161K training examples; we sample 1K for testing). For summarization, we use the TL;DR dataset (Stiennon et al., 2020), which contains annotations of human preference on pairs of summaries (92.9K training examples; we sample 979 for testing). To simulate public data, we use prompts from Alpaca (Taori et al., 2023) (52K instruction-following examples) for the OpenAssistant QA task, SafeRLHF (Ji et al., 2024) (73.9K prompts for safety and performance) for the Anthropic-HH QA task, and Xsum (Narayan et al., 2018) (204K news articles) for the TL;DR summarization task. We evaluate our method in both full public data and limited public data settings, where only 5K prompts are randomly sampled from each dataset.

Setup for Algorithm 1. We use `BAAI/bge-large-en-v1.5` (Xiao et al., 2023) as the embedding model ϕ to encode prompt-response pairs. By default, we set the projected dimension $p = 20$ in DP-PCA and the number of clusters $K = 10$ in K-means. For DP-SGD, we use a learning rate of 0.1, a batch size of 4, 4 training epochs, and a gradient clipping norm of 1.0. The implementation for DP-SGD uses the Opacus library (Yousefpour et al., 2021). To generate candidate responses for public prompts, we use the instruction-finetuned LLaMA-7B-chat model (Touvron et al., 2023) with a temperature of 0.9. We set the number of candidates per prompt to $L = 5$ for the OpenAssistant and Anthropic-HH QA tasks, and $L = 10$ for the TL;DR summarization task. We consider two overall privacy budgets $\varepsilon = 4, 8$. We set $\delta = 1/|\mathcal{D}_{\text{priv}}|$.

Evaluation. For downstream preference learning, we fine-tune the Pythia-2.8B model (Biderman et al., 2023). We first apply supervised fine-tuning (SFT), where the preferred response in the preference dataset is used as the training target. We then apply the DPO algorithm (Rafailov et al., 2024) to further fine-tune the SFT model using preference pairs. Following Rafailov et al. (2024), we measure the win rate of the model-generated responses against the chosen responses in the test set using the GPT-4o model.

Baselines. Since there is no directly comparable work on DP preference learning (see Section 2), we compare our method with two non-private baselines for $\varepsilon = \infty$: (1) fine-tuning directly on the private preference data using SFT and DPO, denoted as $\varepsilon = \infty$ (non syn), and (2) generating synthetic preference data using DPPrefSyn without adding noise, followed by SFT and DPO on the synthetic data, denoted as $\varepsilon = \infty$ (syn). These serve as strong baselines because they represent the best achievable performance without any privacy constraint. We also include a fully private baseline $\varepsilon = 0$, where we evaluate the base model without any preference fine-tuning.

5.1 Main Results

We present our main results in Table 1. We provide the mean and standard deviation of the GPT-4 win rates over 3 runs using different random seeds in DPPrefSyn. In general, our results show that our approach consistently outperforms the strong baseline of direct training on private data ($\varepsilon = \infty$, non syn) across all tasks under $\varepsilon = 4, 8$. Notably, even with only 5K public prompts, it exceeds the $\varepsilon = \infty$ (non syn) baseline on both the OpenAssistant and Anthropic-HH QA tasks. Our approach also achieves performance close to the non-private synthetic baseline $\varepsilon = \infty$ (syn) with $\varepsilon = 4, 8$.

We observe that, our framework consistently achieves higher GPT-4 win rates than the strong baseline of directly fine-tuning on private data ($\varepsilon = \infty$, non syn) across all tasks under $\varepsilon = 4, 8$. For instance, on the Anthropic-HH dataset with $\varepsilon = 4$, our method achieves a 55.32% GPT-4 win rate after SFT and 56.85% after DPO, compared to 31.66% and 38.00% achieved by directly fine-tuning on private data using SFT and DPO, respectively. On the TL;DR summarization task with $\varepsilon = 4$, our approach achieves 61.52% after SFT and 67.89% after DPO, while directly training on private data achieves 25.94% after SFT and 67.31% after DPO. These results highlight the effectiveness of our two-phase framework in enabling high-utility preference learning under strong privacy constraints.

We also observe that using our synthetic data leads to better results after SFT alone, compared to directly perform SFT on private data. For example, on the TL;DR summarization task with $\varepsilon = 4$, our method achieves a GPT-4 win rate of 61.52% after SFT, much higher than the 25.94% achieved

Table 1: Win rates computed by GPT-4o against the chosen responses in the test set. We evaluate our DP framework under $\varepsilon = 4, 8$ using Pythia-2.8B model. Each model is fine-tuned using either: 1) SFT, where the preferred response is used as the training target; or 2) SFT+DPO, where the SFT model is further trained with DPO on preference data. The 5K setting represents using only 5,000 public prompts randomly selected from the full public dataset. $\varepsilon = 0$ denotes the base (fully private, non-finetuned) LLM. We consider two non-private baselines: 1) $\varepsilon = \infty$ (syn) represents our framework without any added noise; 2) $\varepsilon = \infty$ (non syn) represents directly fine-tuning using private dataset. Results show the mean and standard deviation over 3 runs using different random seeds in DPPrefSyn.

Task	Public Prompts	Pref. Learning	$\varepsilon = 0$	$\varepsilon = 4$	$\varepsilon = 8$	$\varepsilon = \infty$ (syn)	$\varepsilon = \infty$ (non syn)
OpenAssistant	Alpaca (5K)	SFT	2.11	$6.93_{1.54}$	$6.23_{0.77}$	$7.49_{0.08}$	4.35
		SFT+DPO		$9.13_{0.37}$	$9.60_{1.41}$	$9.46_{0.43}$	9.13
	Alpaca	SFT		$9.13_{0.48}$	$9.88_{0.70}$	$9.60_{0.58}$	4.35
		SFT+DPO		$12.13_{0.91}$	$12.17_{0.84}$	$12.08_{0.48}$	9.13
Anthropic-HH	SafeRLHF (5K)	SFT	12.14	$49.75_{0.28}$	$49.85_{0.53}$	$48.04_{0.57}$	31.66
		SFT+DPO		$51.72_{1.28}$	$51.69_{2.15}$	$52.06_{0.62}$	38.00
	SafeRLHF	SFT		$55.32_{0.66}$	$55.29_{1.03}$	$55.84_{0.96}$	31.66
		SFT+DPO		$56.85_{2.35}$	$57.18_{1.42}$	$56.96_{0.68}$	38.00
TL;DR	XSum (5K)	SFT	11.64	$45.56_{0.47}$	$45.63_{1.14}$	$48.18_{1.13}$	25.94
		SFT+DPO		$51.89_{1.61}$	$51.62_{0.82}$	$55.09_{1.69}$	67.31
	XSum	SFT		$61.52_{1.94}$	$62.10_{1.61}$	$62.20_{1.41}$	25.94
		SFT+DPO		$67.89_{0.65}$	$71.50_{2.96}$	$71.98_{1.06}$	67.31

by SFT directly on private data. This is because, in our synthesis framework, candidate responses are generated using a high-quality LLM, and only human-aligned responses are selected through private scoring. Therefore, the model receives stronger supervision during SFT, providing a better starting point for subsequent DPO.

We further evaluate our framework in a limited public data setting, using only 5K public prompts for data synthesis. Even with this small amount of public data, our method can outperform the baseline of directly fine-tuning on private data ($\varepsilon = \infty$, non syn). For example, on the Anthropic-HH task with $\varepsilon = 4$, our method achieves a GPT-4 win rate of 51.72% after DPO, outperforming the 38.00% achieved by the $\varepsilon = \infty$ (non syn) baseline, despite using only 5K public prompts, which is about 3% the size of the private dataset. This shows that our framework remains effective even when the available public data is highly constrained. Besides, without any privacy constraint ($\varepsilon = \infty$), using our synthetic data achieves better performance than directly using the real dataset. This represents an effective approach to generating high-quality synthetic preference data, which is of independent interest.

6 Conclusion

In this work, we introduced a two-phase framework for privacy-preserving preference learning, where we generate DP synthetic preference data from private dataset using DPPrefSyn algorithm in phase 1, and use this data to fine-tune LLMs in phase 2. By modeling preference diversity via clustering, improving efficiency with DP-PCA, and conserving privacy budgets through public prompts, DPPrefSyn enables effective and private preference data synthesis. Our experimental results show that our framework achieves strong privacy guarantees while outperforming direct fine-tuning on private data, and remains effective even with limited public data. A potential future direction is to extend our framework to support private learning in multimodal settings, including image and speech.

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.

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. (2023). Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Amin, K., Dick, T., Kulesza, A., Munoz, A., and Vassilvitskii, S. (2019). Differentially private covariance estimation. *Advances in Neural Information Processing Systems*, 32.

Aroyo, L., Diaz, M., Homan, C., Prabhakaran, V., Taylor, A., and Wang, D. (2023a). The reasonable effectiveness of diverse evaluation data. *arXiv preprint arXiv:2301.09406*.

Aroyo, L., Taylor, A., Diaz, M., Homan, C., Parrish, A., Serapio-García, G., Prabhakaran, V., and Wang, D. (2023b). Dices dataset: Diversity in conversational ai evaluation for safety. *Advances in Neural Information Processing Systems*, 36:53330–53342.

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*.

Bakker, M., Chadwick, M., Sheahan, H., Tessler, M., Campbell-Gillingham, L., Balaguer, J., McAleese, N., Glaese, A., Aslanides, J., Botvinick, M., et al. (2022). Fine-tuning language models to find agreement among humans with diverse preferences. *Advances in Neural Information Processing Systems*, 35:38176–38189.

Balle, B. and Wang, Y.-X. (2018). Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In *International Conference on Machine Learning*, pages 394–403. PMLR.

Biderman, S., Schoelkopf, H., Anthony, Q. G., Bradley, H., O'Brien, K., Hallahan, E., Khan, M. A., Purohit, S., Prashanth, U. S., Raff, E., et al. (2023). Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

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

Carranza, A., Farahani, R., Ponomareva, N., Kurakin, A., Jagielski, M., and Nasr, M. (2024). Synthetic query generation for privacy-preserving deep retrieval systems using differentially private language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3920–3930.

Carranza, A. G., Farahani, R., Ponomareva, N., Kurakin, A., Jagielski, M., and Nasr, M. (2023). Privacy-preserving recommender systems with synthetic query generation using differentially private large language models. *arXiv preprint arXiv:2305.05973*.

Chakraborty, S., Qiu, J., Yuan, H., Koppel, A., Manocha, D., Huang, F., Bedi, A., and Wang, M. (2024). Maxmin-rlhf: Alignment with diverse human preferences. In *International Conference on Machine Learning*, pages 6116–6135. PMLR.

Chen, T., Da, L., Zhou, H., Li, P., Zhou, K., Chen, T., and Wei, H. (2024). Privacy-preserving fine-tuning of large language models through flatness. *arXiv preprint arXiv:2403.04124*.

Chowdhury, S. R., Zhou, X., and Natarajan, N. (2024). Differentially private reward estimation with preference feedback. In *International Conference on Artificial Intelligence and Statistics*, pages 4843–4851. PMLR.

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.

Denton, R., Díaz, M., Kivlichan, I., Prabhakaran, V., and Rosen, R. (2021). Whose ground truth? accounting for individual and collective identities underlying dataset annotation. *arXiv preprint arXiv:2112.04554*.

Duan, H., Dziedzic, A., Papernot, N., and Boenisch, F. (2023). Flocks of stochastic parrots: Differentially private prompt learning for large language models. *Advances in Neural Information Processing Systems*, 36:76852–76871.

Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., and Naor, M. (2006a). Our data, ourselves: Privacy via distributed noise generation. In *Advances in Cryptology-EUROCRYPT 2006: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28-June 1, 2006. Proceedings 25*, pages 486–503. Springer.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. (2006b). Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pages 265–284. Springer.

Dwork, C., Roth, A., et al. (2014). The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407.

Gao, F., Zhou, R., Wang, T., Shen, C., and Yang, J. (2024). Data-adaptive differentially private prompt synthesis for in-context learning. *arXiv preprint arXiv:2410.12085*.

Ghazi, B., Golowich, N., Kumar, R., Manurangsi, P., and Zhang, C. (2021). Deep learning with label differential privacy. *Advances in neural information processing systems*, 34:27131–27145.

Gopi, S., Lee, Y. T., and Wutschitz, L. (2021). Numerical composition of differential privacy. *Advances in Neural Information Processing Systems*, 34:11631–11642.

He, J., Li, X., Yu, D., Zhang, H., Kulkarni, J., Lee, Y. T., Backurs, A., Yu, N., and Bian, J. (2022). Exploring the limits of differentially private deep learning with group-wise clipping. *arXiv preprint arXiv:2212.01539*.

Hong, J., Wang, J. T., Zhang, C., Li, Z., Li, B., and Wang, Z. (2023). Dp-opt: Make large language model your privacy-preserving prompt engineer. *arXiv preprint arXiv:2312.03724*.

Hou, C., Shrivastava, A., Zhan, H., Conway, R., Le, T., Sagar, A., Fanti, G., and Lazar, D. (2024). Pre-text: Training language models on private federated data in the age of llms. In *International Conference on Machine Learning*, pages 19043–19061. PMLR.

Jang, J., Kim, S., Lin, B. Y., Wang, Y., Hessel, J., Zettlemoyer, L., Hajishirzi, H., Choi, Y., and Ammanabrolu, P. (2023). Personalized soups: Personalized large language model alignment via post-hoc parameter merging. *arXiv preprint arXiv:2310.11564*.

Ji, J., Hong, D., Zhang, B., Chen, B., Dai, J., Zheng, B., Qiu, T., Li, B., and Yang, Y. (2024). Pku-saferlhf: Towards multi-level safety alignment for llms with human preference. *arXiv preprint arXiv:2406.15513*.

Ji, J., Liu, M., Dai, J., Pan, X., Zhang, C., Bian, C., Chen, B., Sun, R., Wang, Y., and Yang, Y. (2023). Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *Advances in Neural Information Processing Systems*, 36:24678–24704.

Kong, D. and Yang, L. (2022). Provably feedback-efficient reinforcement learning via active reward learning. *Advances in Neural Information Processing Systems*, 35:11063–11078.

Köpf, A., Kilcher, Y., Von Rütte, D., Anagnostidis, S., Tam, Z. R., Stevens, K., Barhoum, A., Nguyen, D., Stanley, O., Nagyfi, R., et al. (2023). Openassistant conversations-democratizing large language model alignment. *Advances in Neural Information Processing Systems*, 36:47669–47681.

Kovač, G., Sawayama, M., Portelas, R., Colas, C., Dominey, P. F., and Oudeyer, P.-Y. (2023). Large language models as superpositions of cultural perspectives. *arXiv preprint arXiv:2307.07870*.

Lee, Y., Williams, J., Marklund, H., Sharma, A., Mitchell, E., Singh, A., and Finn, C. (2024). Test-time alignment via hypothesis reweighting. *arXiv preprint arXiv:2412.08812*.

Li, H., Guo, D., Fan, W., Xu, M., Huang, J., Meng, F., and Song, Y. (2023). Multi-step jailbreaking privacy attacks on chatgpt. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4138–4153.

Li, X., Tramer, F., Liang, P., and Hashimoto, T. (2021). Large language models can be strong differentially private learners. *arXiv preprint arXiv:2110.05679*.

Lin, Z., Gopi, S., Kulkarni, J., Nori, H., and Yekhanin, S. (2023). Differentially private synthetic data via foundation model apis 1: Images. *arXiv preprint arXiv:2305.15560*.

Liu, T., Zhao, Y., Joshi, R., Khalman, M., Saleh, M., Liu, P. J., and Liu, J. (2023). Statistical rejection sampling improves preference optimization. *arXiv preprint arXiv:2309.06657*.

Mattern, J., Jin, Z., Weggenmann, B., Schoelkopf, B., and Sachan, M. (2022). Differentially private language models for secure data sharing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4860–4873.

Mireshghallah, F., Su, Y., Hashimoto, T. B., Eisner, J., and Shin, R. (2023). Privacy-preserving domain adaptation of semantic parsers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4950–4970.

Narayan, S., Cohen, S. B., and Lapata, M. (2018). Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *arXiv preprint arXiv:1808.08745*.

Ochs, S. and Habernal, I. (2024). Private synthetic text generation with diffusion models. *arXiv preprint arXiv:2410.22971*.

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.

Poddar, S., Wan, Y., Ivison, H., Gupta, A., and Jaques, N. (2024). Personalizing reinforcement learning from human feedback with variational preference learning. *arXiv preprint arXiv:2408.10075*.

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.

Rame, A., Couairon, G., Dancette, C., Gaya, J.-B., Shukor, M., Soulier, L., and Cord, M. (2023). Rewarded soups: towards pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. *Advances in Neural Information Processing Systems*, 36:71095–71134.

Saha, A., Pacchiano, A., and Lee, J. (2023). Dueling rl: Reinforcement learning with trajectory preferences. In *International conference on artificial intelligence and statistics*, pages 6263–6289. PMLR.

Sandri, M., Leonardelli, E., Tonelli, S., and Ježek, E. (2023). Why don't you do it right? analysing annotators' disagreement in subjective tasks. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2428–2441.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al. (2024). Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Shen, B., Zhang, J., Chen, T., Zan, D., Geng, B., Fu, A., Zeng, M., Yu, A., Ji, J., Zhao, J., et al. (2023). Pangu-coder2: Boosting large language models for code with ranking feedback. *arXiv preprint arXiv:2307.14936*.

Shoemate, M., Vyrros, A., McCallum, C., Prasad, R., Durbin, P., Casacuberta Puig, S., Cowan, E., Xu, V., Ratliff, Z., Berrios, N., Whitworth, A., Eliot, M., Lebeda, C., Renard, O., and McKay Bowen, C. (2021). OpenDP Library.

Singh, A., Hsu, S., Hsu, K., Mitchell, E., Ermon, S., Hashimoto, T., Sharma, A., and Finn, C. (2025). Fspo: Few-shot preference optimization of synthetic preference data in llms elicits effective personalization to real users. *arXiv preprint arXiv:2502.19312*.

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.

Tan, B., Xu, Z., Xing, E., Hu, Z., and Wu, S. (2025). Synthesizing privacy-preserving text data via finetuning without finetuning billion-scale llms. *arXiv preprint arXiv:2503.12347*.

Tang, X., Panda, A., Nasr, M., Mahloujifar, S., and Mittal, P. (2024). Private fine-tuning of large language models with zeroth-order optimization. *arXiv preprint arXiv:2401.04343*.

Tang, X., Shin, R., Inan, H. A., Manoel, A., Mireshghallah, F., Lin, Z., Gopi, S., Kulkarni, J., and Sim, R. (2023). Privacy-preserving in-context learning with differentially private few-shot generation. *arXiv preprint arXiv:2309.11765*.

Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., and Hashimoto, T. B. (2023). Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, 3(6):7.

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*.

Vogels, E. A. (2021). *The state of online harassment*, volume 13. Pew Research Center Washington, DC.

Wang, W., Liang, X., Ye, R., Chai, J., Chen, S., and Wang, Y. (2024). Knowledgesg: Privacy-preserving synthetic text generation with knowledge distillation from server. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7677–7695.

Wu, F., Inan, H. A., Backurs, A., Chandrasekaran, V., Kulkarni, J., and Sim, R. (2023a). Privately aligning language models with reinforcement learning. *arXiv preprint arXiv:2310.16960*.

Wu, S., Xu, Z., Zhang, Y., Zhang, Y., and Ramage, D. (2024). Prompt public large language models to synthesize data for private on-device applications. *arXiv preprint arXiv:2404.04360*.

Wu, T., Panda, A., Wang, J. T., and Mittal, P. (2023b). Privacy-preserving in-context learning for large language models. *arXiv e-prints*, pages arXiv–2305.

Xiao, S., Liu, Z., Zhang, P., and Muennighoff, N. (2023). C-pack: Packaged resources to advance general chinese embedding.

Xie, C., Lin, Z., Backurs, A., Gopi, S., Yu, D., Inan, H. A., Nori, H., Jiang, H., Zhang, H., Lee, Y. T., et al. (2024). Differentially private synthetic data via foundation model apis 2: Text. In *International Conference on Machine Learning*, pages 54531–54560. PMLR.

Yousefpour, A., Shilov, I., Sablayrolles, A., Testuggine, D., Prasad, K., Malek, M., Nguyen, J., Ghosh, S., Bharadwaj, A., Zhao, J., et al. (2021). Opacus: User-friendly differential privacy library in pytorch. *arXiv preprint arXiv:2109.12298*.

Yu, D., Kairouz, P., Oh, S., and Xu, Z. (2024). Privacy-preserving instructions for aligning large language models. In *International Conference on Machine Learning*, pages 57480–57506. PMLR.

Yu, D., Naik, S., Backurs, A., Gopi, S., Inan, H. A., Kamath, G., Kulkarni, J., Lee, Y. T., Manoel, A., Wutschitz, L., et al. (2021). Differentially private fine-tuning of language models. *arXiv preprint arXiv:2110.06500*.

Yue, X., Inan, H., Li, X., Kumar, G., McAnallen, J., Shajari, H., Sun, H., Levitan, D., and Sim, R. (2023). Synthetic text generation with differential privacy: A simple and practical recipe. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1321–1342.

Zhang, J., Lei, M., Ding, M., Li, M., Xiang, Z., Xu, D., Xu, J., and Wang, D. (2025). Towards user-level private reinforcement learning with human feedback. *arXiv preprint arXiv:2502.17515*.

Zhang, L., Thekumparampil, K. K., Oh, S., and He, N. (2023). Dpzero: Dimension-independent and differentially private zeroth-order optimization. *arXiv preprint arXiv:2310.09639*.

Zhu, B., Jordan, M., and Jiao, J. (2023). Principled reinforcement learning with human feedback from pairwise or k-wise comparisons. In *International Conference on Machine Learning*, pages 43037–43067. PMLR.

Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., and Irving, G. (2019). Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

A Privacy Analysis

In this section, we provide the privacy analysis for Algorithm 1.

Theorem 2 (Restatement of Theorem 1)**.** *Algorithm 1 is (ε, δ)-differentially private.*

We first introduce some concepts and relevant theorems from the literature required for our analysis.

Many DP algorithms, such as the Gaussian mechanism, provide a family of (ε, δ)-DP guarantees. Specifically, for each fixed ε, there exists a $\delta(\varepsilon)$, such that the mechanism satisfies $(\varepsilon, \delta(\varepsilon))$-DP.

Definition 2 (Privacy curve)**.** A DP algorithm $\mathcal{M}$ is said to have a *privacy curve* $\delta : \mathbb{R} \to [0, 1]$ if, for every $\varepsilon > 0$, the algorithm $\mathcal{M}$ satisfies $(\varepsilon, \delta(\varepsilon))$-DP.

The following result on the privacy curve of Gaussian mechanism is due to Balle and Wang (2018).

Theorem 3 (Privacy curve of Gaussian mechanism (Balle and Wang, 2018))**.** *Let $f : \mathcal{X} \to \mathbb{R}$ be a function with global ℓ_2-sensitivity Δ. For any $\varepsilon > 0$ and $\delta \in [0, 1]$, the Gaussian output perturbation mechanism $M(x) = f(x) + Z$ with $Z \sim \mathcal{N}(0, \sigma^2 I)$ is (ε, δ)-DP if and only if*

$$\Phi\left(\frac{\Delta}{2\sigma} - \frac{\varepsilon\sigma}{\Delta}\right) - e^{\varepsilon} \cdot \Phi\left(-\frac{\Delta}{2\sigma} - \frac{\varepsilon\sigma}{\Delta}\right) \leq \delta.$$

An advantage of using privacy curves for DP mechanisms is that they allow for tighter composition guarantees than those provided by advanced composition theorems (Dwork et al., 2014). Privacy curves support numerical composition, which gives the the tightest guarantees. Gopi et al. (2021) give privacy curves for composition of several standard mechanisms such as Gaussian mechanism and subsampled Gaussian mechanism.

Theorem 4 (Gopi et al. (2021))**.** *Suppose $M_1, M_2, \ldots, M_k$ are DP algorithms. Then the privacy curve $\delta_M(\varepsilon)$ of adaptive composition $M = M_1 \circ M_2 \circ \cdots \circ M_k$ can be approximated in time*

$$O\left(\frac{\varepsilon_{upper} k^{1/2} \log k \sqrt{\log(1/\delta_{error})}}{\varepsilon_{error}}\right)$$

where ε_{error} is the additive error in ε, δ_{error} is the additive error in δ, and ε_{upper} is an upper bound on

$$\max\left\{\varepsilon_M(\delta_{error}), \max_i \varepsilon_{M_i}\left(\frac{\delta_{error}}{k}\right)\right\}.$$

Gopi et al. (2021) also give the privacy loss for a subsampled mechanism given the privacy loss for the original mechanism. This can be used to bound the privacy loss of DP-SGD.

Theorem 5 (Gopi et al. (2021))**.** *Let (X, Y) be the PRVs for a privacy curve $\delta(P \parallel Q)$. Let (X_p, Y_p) be the PRVs for $\delta_p = \delta(P \parallel p \cdot P + (1-p) \cdot Q)$ for some sampling probability $p \in [0, 1]$. Then*

$$X_p = \log(1 + p(e^X - 1)), \quad Y_p = \begin{cases} \log(1 + p(e^Y - 1)) & \text{w.p. } p \\ \log(1 + p(e^X - 1)) & \text{w.p. } 1 - p. \end{cases}$$

The CDFs of X_p and Y_p are given by:

$$\mathrm{CDF}_{X_p}(t) = \begin{cases} \mathrm{CDF}_X\left(\log\left(\frac{e^t - (1-p)}{p}\right)\right) & \text{if } t \geq \log(1-p) \\ 0 & \text{if } t < \log(1-p). \end{cases}$$

$$\mathrm{CDF}_{Y_p}(t) = \begin{cases} p \cdot \mathrm{CDF}_Y\left(\log\left(\frac{e^t - (1-p)}{p}\right)\right) + (1-p) \cdot \mathrm{CDF}_X\left(\log\left(\frac{e^t - (1-p)}{p}\right)\right) & \text{if } t \geq \log(1-p) \\ 0 & \text{if } t < \log(1-p). \end{cases}$$

We are ready to do the privacy analysis of our algorithm (Theorem 1).

Proof. Our algorithm consists of 3 components that access the private dataset: a DP-PCA subroutine that satisfies ε_0-DP (Amin et al., 2019) in Line 5, DP-SGD with noise multiplier σ_1 in Line 9 and DP histogram with noise multiplier σ_2 in Line 12. We allocate $\varepsilon_0 = 0.5$ for total budget $\varepsilon = 4$ and

$\varepsilon_0 = 1.0$ for $\varepsilon = 8$. The remaining budget is allocated to DP-SGD and DP histogram, composed using the Privacy Random Variable accountant (Gopi et al., 2021).

In Line 7, we split low-dimensional embeddings $\{z_i\}_{i=1}^n$ into K disjoint groups $\mathcal{C}_1, ..., \mathcal{C}_K$. A separate linear model is trained on each cluster using DP-SGD in Line 9. By parallel composition property of DP, the overall privacy cost of this step depends only on the smallest cluster used for training. To ensure a known lower bound on the sample size in each model, we discard clusters with fewer than $\mathcal{D}_{\text{priv}}/(K+3)$ samples. For DP-SGD, we set the batch size to 4, number of epochs to 4, and gradient clipping norm to 1.0. The per-iteration privacy loss in DP-SGD follows the privacy curve of the subsampled Gaussian mechanism (Theorem 5).

In Line 11, we compute a histogram vector $\mathbf{h} \leftarrow [|\mathcal{C}_1|, \ldots, |\mathcal{C}_K|]$. The ℓ_2 sensitivity of the term $\mathbf{h}$ is at most $\sqrt{2}$. As we add Gaussian noise $\mathcal{N}(0, \sigma_2^2 \mathbb{I})$ to it, this step has the same privacy curve as given in Theorem 3. We then appeal to Theorem 4 for composing DP-SGD and DP histogram (Gopi et al., 2021), making sure that it satisfies $(\varepsilon - \varepsilon_0, \delta)$-DP.

We provide the codes for privacy accounting below.

```
def get_privacy_spent(sampling_prob_dpsgd, running_steps_dpsgd,
                      noise_multiplier_dpsgd, noise_multiplier_histogram, delta):

    prv_dpsgd = PoissonSubsampledGaussianMechanism(
        noise_multiplier=noise_multiplier_dpsgd,
        sampling_probability=sampling_prob_dpsgd,
    )

    prv_histogram = GaussianMechanism(
        noise_multiplier=noise_multiplier_histogram,
        l2_sensitivity = np.sqrt(2),
    )

    accountant = PRVAccountant(
        prvs=[prv_dpsgd, prv_histogram],
        max_self_compositions=[running_steps_dpsgd, 1],
        eps_error=0.01,
        delta_error=delta/10,
    )

    eps_lower, eps_estimate, eps_upper = accountant.compute_epsilon(
        delta=delta,
        num_self_compositions=[running_steps_dpsgd, 1],
    )

    return eps_upper
```

□