Privacy-Preserving Instructions for Aligning Large Language Models

Da Yu[†] **Peter Kairouz**[‡] **Sewoong Oh**[‡] **Zheng Xu**[‡]

Abstract

Service providers of large language model (LLM) applications collect user instructions in the wild and use them in further aligning LLMs with users' intentions. These instructions, which potentially contain sensitive information, are annotated by human workers in the process. This poses a new privacy risk not addressed by the typical private optimization. To this end, we propose using synthetic instructions to replace real instructions in data annotation and model fine-tuning. Formal differential privacy is guaranteed by generating those synthetic instructions using privately fine-tuned generators. Crucial in achieving the desired utility is our novel filtering algorithm that matches the distribution of the synthetic instructions to that of the real ones. In both supervised fine-tuning and reinforcement learning from human feedback, our extensive experiments demonstrate the high utility of the final set of synthetic instructions by showing comparable results to real instructions. In supervised fine-tuning, models trained with private synthetic instructions outperform leading open-source models such as Vicuna.

1. Introduction

Training models to follow instructions is fundamental to the development of large language model (LLM) applications (Wei et al., 2021; Chung et al., 2022; Ouyang et al., 2022). Instructions in the form of prompts provided by users during interactions with LLM applications are used to refine the backbone models (OpenAI, 2023). These instructions are often reviewed and annotated by human annotators to create the training data for this refinement process (Tan & Cabato, 2023). However, user-provided instructions can be highly sensitive since users may disclose personal information to elicit accurate responses. To illustrate this, we conducted a quantitative analysis on approximately twelve thousand real-world user instructions collected from the Chatbot Arena (Zheng et al., 2023b). This analysis begins with an LLM-based detection, followed by human verification, as detailed in Appendix B. Our analysis finds at least thousand of sensitive text pieces. A few examples of sensitive instructions are shown in Figure 2.

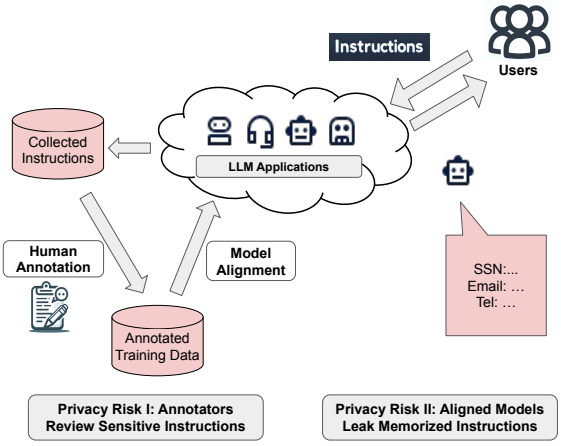


Figure 1. Privacy concerns of collecting and training with user instructions in LLM applications.

Given the sensitive nature of user instructions, we identify the following privacy risks in the pipeline of aligning LLMs with instructions, as illustrated in Figure 1. During the data collection stage, potentially sensitive instructions are collected and presented to human annotators. In the subsequent training and deployment stages, trained models may memorize training data, including sensitive instructions, and inadvertently leak this memorized data during deployment. While the privacy risk of model memorization has been discussed in previous studies (Carlini et al., 2021; 2023; Nasr et al., 2023a; Wu et al., 2024a), to the best of our knowledge, this work is the first to discuss the privacy risks associated with the annotation process.

One common approach to mitigate privacy concerns is to

[†]Sun Yat-sen University, the work was done when Da Yu was an intern at Google Research. yuda3@mail2.sysu.edu.cn.
[‡]Google Research, alphabetical order.
{kairouz,sewoongo,xuzheng}@google.com.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

detect sensitive information in user instructions by either human annotation or automatic tools, and then remove this information from the model training process. For example, Ouyang et al. (2022) ask annotators to remove personally identifiable information (PII) from instructions before using them for model training (OpenAI, 2022, Footnote A). However, there are several concerns that cannot be fully addressed by PII removal alone. First, detecting PII by human annotators exposes sensitive instructions to annotators. Second, PII detection by automatic tools has high false negative rates. Bubeck et al. (2023) show that GPT-4, despite far surpassing existing automatic tools for PII detection, cannot detect all pieces of personal information in over 20% of the documents in the Text Anonymization Benchmark (Pilán et al., 2022). Third, content that appears safe and cannot be identified as PII during the detection process may still reveal a user's identity if the adversary has side information (Narayanan & Shmatikov, 2008). To address these limitations, we develop a framework for aligning LLMs with instructions while providing differential privacy (DP) guarantees (Dwork et al., 2006b), which theoretically limit the influence of any individual training sample and offer strong protection against empirical attacks (Rahman et al., 2018; Carlini et al., 2019; Lowy et al., 2024).

The standard method for fine-tuning LLMs on private data is to replace standard optimizers with their DP variants (Yu et al., 2021; Li et al., 2022b; Bu et al., 2022; Wu et al., 2024a). However, merely ensuring that the fine-tuned model satisfies DP can only address the risk of memorization; there is no protection in the data collection stage when the user instructions are exposed to annotators. To simultaneously address both privacy risks, human annotators and model memorization, we propose using synthetic instructions (generated from privately fine-tuned language models) as substitutes for real user instructions in the alignment pipeline.

Contributions. We make the following contributions.

We introduce a two-stage framework for privately generating high quality synthetic instructions, which can be seamlessly incorporated into the training pipeline of instruction-following models (Figure 3). First, an LLM pre-trained on public data is privately fine-tuned on user-provided instructions, from which a large set of synthetic instructions are sampled (Algorithm 1). We adopt state-of-the-art strategies for DP fine-tuning including initializing with a large pre-trained model and parameter-efficient fine-tuning (Yu et al., 2022; Yue et al., 2022; Kurakin et al., 2023).

There is still a large distribution shift between the user-provided instructions and the initial set of private machine-generated instructions. To bridge this gap, we introduce the second stage of a novel private resampling algorithm (Algorithm 2); we select a subset of the initial private synthetic instructions in order to better match the real user instructions. Specifically, we first use a private histogram to approximate the distribution of the real instruction dataset, where each bin corresponds to each cluster learned from synthetic instructions, for example, representing coding questions for a programming language. The private histogram is then used to filter the initial synthetic instructions. The construction of the histogram is inspired by Pillutla et al. (2022) which uses similar cluster-and-histogram strategy to measure divergences between two datasets.

We demonstrate the performance of the proposed approach on standard methods of aligning language models: for supervised instruction fine-tuning on LLaMA (Touvron et al., 2023a) 7B and 13B models and for reinforcement learning from human feedback on 1.3B Phi-1.5 model (Li et al., 2023b). We evaluate them on instructions from the LMSYS-Chat-1M dataset (Zheng et al., 2023a) and the AlpacaEval benchmark (Dubois et al., 2023). In supervised fine-tuning, training a 7B LLaMA model with resampled DP synthetic instructions leads to an 8.6% relative improvement compared to using the unfiltered initial DP synthetic instructions. The improvement is measured in the win-rate against a baseline model. In reinforcement learning from human feedback, our private approach is comparable to using real instructions without any privacy guarantees.

Overview. We provide background on fine-tuning LLMs on instructions in Section 2 and discuss privacy risks involved. A new framework for generating high-quality synthetic instructions with DP is presented in Section 3. Empirical results demonstrating the utility of those synthetic instructions are provided in Section 4. We end with a conclusion in Section 5.

2. Privacy Risks and Background

We present necessary background and closely related prior work. Additional related work is discussed in Appendix A.

Privacy risks in training LLMs to follow instructions. Aligning LLMs with user instructions is crucial for ensuring accurate and contextually relevant responses, which in turn significantly enhances the user experience (Ziegler et al., 2019; Ouyang et al., 2022). When aligning pre-trained LLMs, a combination of the following two popular methods is used: (i) supervised fine-tuning, which utilizes (instruction, answer) pairs to minimize the cross-entropy loss, and (ii) reinforcement learning from human feedback, where a reward model is trained on human preferences in the form of (instruction, {answer}, preference) to evaluate and score answers. Real-world user instructions are necessary in both alignment techniques, which might contain sensitive personal information. However, these instructions are sent to human annotators to get the answers in supervised fine-tuning and to get preferences in reward modeling (Ouyang

[Type: creative writing.] You are a grant writing super hero. Write up a basic grant for [Organization Name] in [Location], to get 25,000 dollars for a new [Event Name].

[Type: information extraction.] [Medical Report]
can you find the Date of servvice and provider name

[Type: brainstorming.] I am a [Job Title] working in a [Country Name] company that ..... At the same time, since [Date], I follow studies at [Program Name]. ..... I'd like the AI to support specifically in defining key questions to .....

[Type: document rewriting.] i need you to act as a professional human translator, native speaker in English Italian ..... the following is a business email, translate it in idiomatic italian with a formal tone: Hello xxx, I am xxx, [Job Title] of [Organization Name]. I am contacting you to see if you would be interested in bringing your creative vision to [Location]. Based in [Location], we are .....

Figure 2. Samples of instructions containing personal information. We mask the sensitive texts to protect user privacy.

et al., 2022; Touvron et al., 2023b). This introduces privacy concerns referred to as Privacy Risk I in Figure 1. The standard practice of keeping the alignment dataset proprietary is not sufficient to preserve the privacy of the participants.

Only recently has it become possible to identify this vulnerability, following the release of several datasets featuring real-world user interactions with instruction-following models (Zheng et al., 2023a; Zhao et al., 2024). We focus on two such datasets collected from the Chatbot Arena (Zheng et al., 2023a;b), carefully adhering to their terms of use, and study privacy implications of using real-world instructions.

For the Chatbot Arena Conversations dataset (Zheng et al., 2023b), the authors mention that "To ensure the safe release of data, we have made our best efforts to remove all conversations that contain personally identifiable information (PII)." Despite that, we identified numerous sensitive pieces of information in the publicly available dataset, which demonstrates how unreliable PII detection is and how vulnerable the instruction datasets are. The detailed PII analysis of the Chatbot Arena Conversations dataset is provided in Appendix B and sample instructions containing personal information are shown in Figure 2.

Private deep learning. We use (ε,δ)-DP (Dwork et al., 2006a) defined as,

Definition 2.1 ((ε, δ)-Differential Privacy)**.** A randomized algorithm $\mathcal{M}$ satisfies (ε,δ)-DP for $\mathbb{D}$ if for any two neighboring datasets $\mathbb{D}$, $\mathbb{D}'$ and for all $\mathcal{S} \subset \text{Range}(\mathcal{M})$:

$$\Pr[\mathcal{M}(\mathbb{D}) \in \mathcal{S}] \leq e^{\varepsilon} \Pr[\mathcal{M}(\mathbb{D}') \in \mathcal{S}] + \delta.$$

Neighboring datasets are defined by adding/removing one sample, where each sample constitutes a single user instruction (text prompt), i.e., example-level DP (Ponomareva et al., 2023, Section 5.1). $\mathcal{M}$ is a (randomized) training algorithm that outputs trained model weights. The most common approach for private deep learning is to clip and add noise to each gradient update (Song et al., 2013; Abadi et al., 2016; Kairouz et al., 2021). We use a variant of DP-SGD (Abadi et al., 2016) with Adam optimizer, which is called DP-Adam (Li et al., 2022b).

Differentially private synthetic data. A widely adopted method for generating private synthetic text in practice is to train a language model on private data with DP, and then repeatedly sample the DP model to generate synthetic text sequences (Augenstein et al., 2019; Bommasani et al., 2019; Putta et al., 2022; Mattern et al., 2022; Yue et al., 2022; Kurakin et al., 2023). In this work, we follow this general approach to generate initial synthetic instructions. Importantly, we extend beyond the scope of prior studies, which primarily focus on simple tasks such as text classification, by investigating the use of private synthetic text in the training of instruction-following models. Moreover, we introduce a novel filtering algorithm that substantially reduces the distributional gap between real and private synthetic texts.

Privacy-preserving In-Context Learning (ICL). In addition to aligning LLMs through fine-tuning, ICL is another paradigm for adapting LLMs to specific domains. ICL improves a model's performance in a target domain by providing instructions and few-shot demonstrations directly in the prompt (Brown et al., 2020). To protect the privacy of in-context prompts (Duan et al., 2023b), a recent line of research focuses on generating differentially private prompts for in-context learning (Duan et al., 2024; Wu et al., 2024b; Tang et al., 2024b; Hong et al., 2024). Our work differs from DP-ICL in two key aspects. Firstly, regarding the threat model, DP-ICL protects sensitive information within in-context prompts, while our approach protects the privacy of training data against both model memorization and exposure to human annotation. Secondly, in terms of algorithms, our framework produces a large set of synthetic instructions rather than few-shot demonstrations, catering to the requirement of a larger set of data in fine-tuning for LLM alignment.

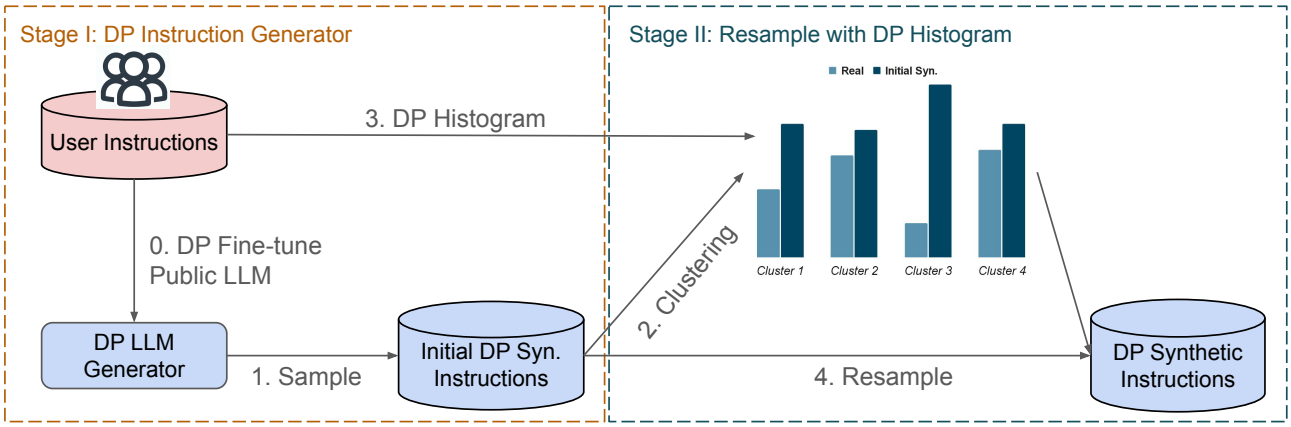


Figure 3. Our two-stage framework for privately generating high-quality synthetic instructions.

3. Generating Synthetic Instructions with Differential Privacy

To mitigate the privacy concerns in aligning LLMs, we propose a novel privacy-preserving approach by generating synthetic instructions with DP (see Figure 3 for an overview). First, an LLM generator is fine-tuned with DP on user-provided instructions, which is subsequently used to generate a large set of initial *synthetic* instructions (Algorithm 1). Next, these initial synthetic instructions are resampled according to a private histogram to approximately match the distribution of the real instructions (Algorithm 2). We carefully allocate the privacy budget in Algorithms 1 and 2, and the formal DP guarantees of the entire process can be composed (Kairouz et al., 2015; Steinke, 2022). The generated DP synthetic data can be used for aligning LLMs and the same DP guarantees are applied because of the post-processing property of DP (Dwork et al., 2014).

3.1. Stage 1: DP Instruction Generator

We take an LLM, pre-trained on public data, and fine-tune on private instruction data with DP. This is used to generate samples that comprise a large set of initial instruction dataset, see Algorithm 1. Concretely, LLaMA models (Touvron et al., 2023a) are privately fine-tuned on user instructions with LoRA (Hu et al., 2022). LoRA has been shown to have superior privacy-utility trade-offs in fine-tuning of DP text generators (Kurakin et al., 2023). Previous research has shown that DP training is effective at preventing memorization of training data (Guo et al., 2022; Stock et al., 2022; Yue et al., 2022). In Appendix C, we provide further evidence by by showing that text generative models trained without DP can memorize training samples and leak them during inference, whereas DP-trained models do not.

Since the fine-tuned model itself is private, we can sample as many instructions as we want. However, there is a non-negligible distributional gap between the initial synthetic instructions and real instructions, which increases as the DP guarantee strengthens, likely due to the noise added in DP training. The results in Table 1 in Section 4.2 illustrate this phenomenon. We then present a DP resampling algorithm to improve the quality of initial synthetic instructions.

Algorithm 1 Train DP Generator to Synthesize Instructions.

1: **Input:** real instructions $\mathbb{D}_{real}$, pre-trained model θ, privacy parameters $(\varepsilon_1, \delta_1)$, number of generations M.
2: Fine-tune θ on $\mathbb{D}$ with DP-Adam and $(\varepsilon_1, \delta_1)$-DP.
3: Sample from the fine-tuned model for M times to generate M synthetic instructions ($\mathbb{D}_{syn}$).
4: **return** $\mathbb{D}_{syn}$

3.2. Stage 2: Resample with DP Histogram

Let $\mathbb{D}_{real}$ be a set of real instructions of size N, $\mathbb{D}_{syn}$ be a set of initial synthetic instructions of size M ($M > N$), and ϕ be an embedding model that maps an instruction into a real-valued vector. Our goal is to select $\mathbb{D}'_{syn} \subseteq \mathbb{D}_{syn}$ whose distribution more closely aligns with that of $\mathbb{D}_{real}$ in the embedding space.

The pseudocode of our resampling process is presented in Algorithm 2. We first groups synthetic instructions, $\mathbb{D}_{syn}$, into K clusters, $\{\boldsymbol{c}_j\}_{j=1}^{K}$, using k-means. Clustering synthetic instructions does not incur additional privacy cost because it constitutes a post-processing step of Algorithm 1. The cluster centroids of $\mathbb{D}_{syn}$ are used to create a histogram of real instruction samples in $\mathbb{D}_{real}$ where each bin corresponds to each cluster. This histogram, $\boldsymbol{h}$, is made private by adding appropriate Gaussian noise to each bin. The synthetic data, $\mathbb{D}_{syn}$, is resampled to match this private histogram. Table 1 in Section 4.2 demonstrates that the filtered $\mathbb{D}'_{syn}$ is closer to $\mathbb{D}_{real}$ than the initial $\mathbb{D}_{syn}$, which in turn

Algorithm 2 Resample Synthetic Data with DP Histogram.

Input: initial synthetic instructions $\mathbb{D}_{syn}$, real instructions $\mathbb{D}_{real}$, embedding model ϕ, number of clusters K, privacy budget $(\varepsilon_2, \delta_2)$-DP, number of target synthetic samples T.
Compute the embeddings of synthetic instructions $\{\boldsymbol{e}_i^{syn}\}_{i=1}^M$ and real instructions $\{\boldsymbol{e}_i^{real}\}_{i=1}^N$ with ϕ.
Run k-means to cluster $\{\boldsymbol{e}_i^{syn}\}$ into K groups. Let the i_{th} group be $\mathbb{G}_i$ and the corresponding centroid be $\boldsymbol{c}_i$.
Initialize a K-dimensional histogram $\boldsymbol{h}$.
//Each real instruction chooses the nearest centroid.
for $i = 1$ **to** $|\mathbb{D}_{real}|$ **do**
  $\gamma = \arg\min_{j=1}^{K} |\boldsymbol{e}_i^{real} - \boldsymbol{c}_j|$.
  $\boldsymbol{h}[\gamma] = \boldsymbol{h}[\gamma] + 1$.
end for
//Privatize the histogram and compute the densities.
$\tilde{\boldsymbol{h}} = \boldsymbol{h} + z, z \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I}_{K\times K})$, where σ is set to guarantee $(\varepsilon_2, \delta_2)$-DP.
$\boldsymbol{p} = \tilde{\boldsymbol{h}}/|\mathbb{D}_{real}|$.
//Resample initial synthetic instructions.
for $i = 1$ **to** K **do**
  if $|\mathbb{G}_i| < \lceil T * \boldsymbol{p} \rceil$ **then**
    return *'Need more initial samples.'*
  else
    Uniformly sample $\max(\lceil T * \boldsymbol{p} \rceil, 0)$ synthetic instructions from $\mathbb{G}_i$. Let $\hat{\mathbb{G}}_i$ be the sampled subset.
  end if
end for
return $\cup_{i=1;K} \hat{\mathbb{G}}_i$.

improves the performance of the model fine-tuned on the filtered dataset. Further studying the clusters of $\mathbb{D}_{syn}$ reveals that, for example, the top voted cluster contains instructions of coding questions in Python as shown in Figure 17 of Appendix G. On the other hand, clusters receive least votes do not exhibit any coherent semantics.

Algorithm 2 reduces the gap between real and synthetic datasets by aligning their histograms. The idea of clustering samples in a dataset into a histogram is inspired by MAUVE (Pillutla et al., 2021; 2022). MAUVE is a metric designed to quantify the discrepancy between synthetic and real data. It first selects methods to represent the distributions of both synthetic and real datasets and then measures the divergence frontiers (Djolonga et al., 2020). One specific method involves clustering the samples in a dataset into a histogram and using this histogram as the approximated distribution.

The size of the histogram, K, traverses a trade-off between signal and noise: a large K gives a fine-grained characterization of the real instruction distribution, but also makes the histogram harder to privatize because of the increased dimension and reduced votes in each cluster. As the votes

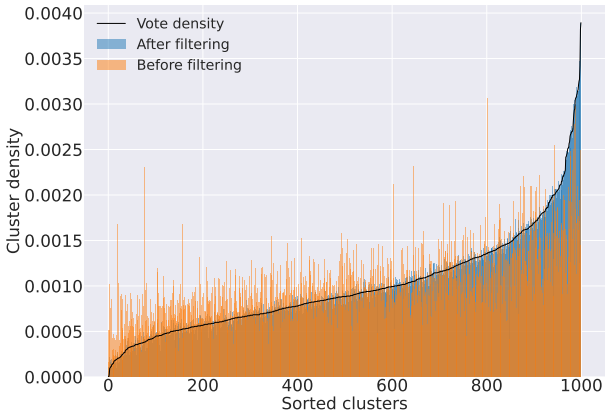


Figure 4. Probability densities of clusters of synthetic instructions. The black line shows the sorted votes from real samples (before noising). The filtering process aligns the distribution of synthetic instructions with that of real instructions.

from real instructions sum up to N, a large K reduces the number of votes per-cluster. Therefore, adding proper noise to the histogram for meaningful DP guarantees would result in an inaccurate estimation of the real instruction distribution due to poor signal-to-noise ratio. One good practice for determining the value of K is to ensure that the average number of votes per cluster is sufficiently robust to DP noise. In our experiments, we investigate the quality of resampled synthetic data, measured by MAUVE, for varying K. As depicted in Figure 5 of Section 4.2, while the resampling process enhances the quality of synthetic data for all choices of K, the improvement initially increases with K but beyond a certain point, it begins to either plateau or decline.

We illustrate the resample process in Figure 4 with clusters $K = 1000$ and noise $\sigma = 10$. We fine-tune LLaMA 13B (Touvron et al., 2023a) with DP-Adam and sample one million synthetic instructions from the fine-tuned model. The overall privacy cost for fine-tuning and resampling is $(5.98, 5 \times 10^{-7})$-DP. Other implementation details, such as the choice of the embedding model, are in Section 4. After filtering, there are around 310 thousand synthetic instructions left. Figure 4 shows that the empirical clustered distribution of synthetic instructions after filtering matches that of real instructions.

It is worth noting that Algorithm 2 has a much lower privacy cost compared to Algorithm 1. It only releases a histogram once which can tolerate much larger noise. For example, as depicted in Table 1 of Section 4.2, selecting $\varepsilon_1 = 5.94$ and ε_2 small enough that the end-to-end privacy is only $\varepsilon = 5.98$ is already enough to see significant gain of filtering.

4. Experiments

We conduct extensive experiments to evaluate our algorithms. In Section 4.2, we assess the distributional gap between synthetic and real instructions. We then investigate the utility of synthetic instructions in both supervised instruction fine-tuning (Section 4.3) and reinforcement learning from human feedback (Section 4.4).

4.1. Setup for Generating Synthetic Instructions

Instruction datasets. We use the LMSYS-Chat-1M dataset as the private dataset (Zheng et al., 2023a), which has one million conversations between real users and instruction-following models collected from April to August 2023. We only use the user instructions and disregard the corresponding machine-generated responses. This data is first preprocessed by deduplication and filtering out non-English instructions, among other steps, as detailed in Appendix E. After preprocessing, approximately 200,000 instructions remain, which we refer to as Chatbot Arena instructions. The Chatbot Arena instructions are then divided into three subsets: 180,000 for the training set, 5,000 for the validation set, and the rest for the test set.

We compare the models fine-tuned on private instructions (with DP) with models fine-tuned on out-of-distribution public instructions (without DP). This comparison aims to demonstrate that, despite the abundance of public instruction datasets (Longpre et al., 2023; Lian et al., 2023), in-domain but sensitive user instructions are still essential for aligning LLMs. For this, we take instructions from the FLAN dataset (Chung et al., 2022; Longpre et al., 2023). FLAN comprises question/answer pairs derived from a rich collection of classic NLP tasks. For our purposes, we retain only the instructions. We randomly sample 180,000 instructions from FLAN unless otherwise specified.

Privacy accounting. We consider two overall privacy budgets $\varepsilon \simeq 3$ and $\varepsilon \simeq 6$. The privacy parameter δ is set as 5×10^{-7}, smaller than $0.1/N$, where N=180,000 is the number of training samples. We use the Privacy Random Variable accountant (Koskela et al., 2020; Gopi et al., 2021; Ghazi et al., 2022) to compose the privacy costs of Algorithm 1 and 2. The code snippet for privacy accounting is in Appendix F.4.

Setup for Algorithm 1. The DP generators are fine-tuned from LLaMA models (Touvron et al., 2023a). This ensures that no private data is leaked in the pre-training, since the cutoff date for the pre-training data of LLaMA is prior to the start of the collection of Chatbot Arena instructions. We tune the hyperparameters based on MAUVE scores. Details regarding the hyperparameters for both fine-tuning and sampling are in Appendix F.2.

We undertake several measures to generate high-quality initial synthetic instructions. Using MAUVE score between Chatbot Arena instructions and the initial synthetic instructions, we study the effect of model size with LLaMA 7B and 13B models. The results suggest that a larger model synthesizes better quality data (Appendix D.2). One unexpected by-product of gradient clipping in DP fine-tuning is that the clipping threshold has a significant impact on the length of synthetic instructions and, consequently, affects the MAUVE scores. The results of using varying clipping thresholds are in Appendix D.1. In the rest of this paper, generator models are fine-tuned from LLaMA 13B unless otherwise specified. We sample one million initial synthetic instructions from each fine-tuned model.

Setup for Algorithm 2. We cluster instruction datasets using K-means clustering on the Sentence-T5-base (Ni et al., 2021) embeddings, using the Faiss package for an efficient implementation (Johnson et al., 2019). Unless otherwise specified, the number of clustering bins K and noise standard deviation σ in Algorithm 2 are set as 1000 and 10, respectively. Selecting a subset of 180,000 synthetic instructions requires around 743,000 and 574,000 initial synthetic samples for $\varepsilon \simeq 3$ and $\varepsilon \simeq 6$, respectively. A larger number of initial samples are required for higher privacy, i.e., $\varepsilon \simeq 3$, since sample quality is lower.

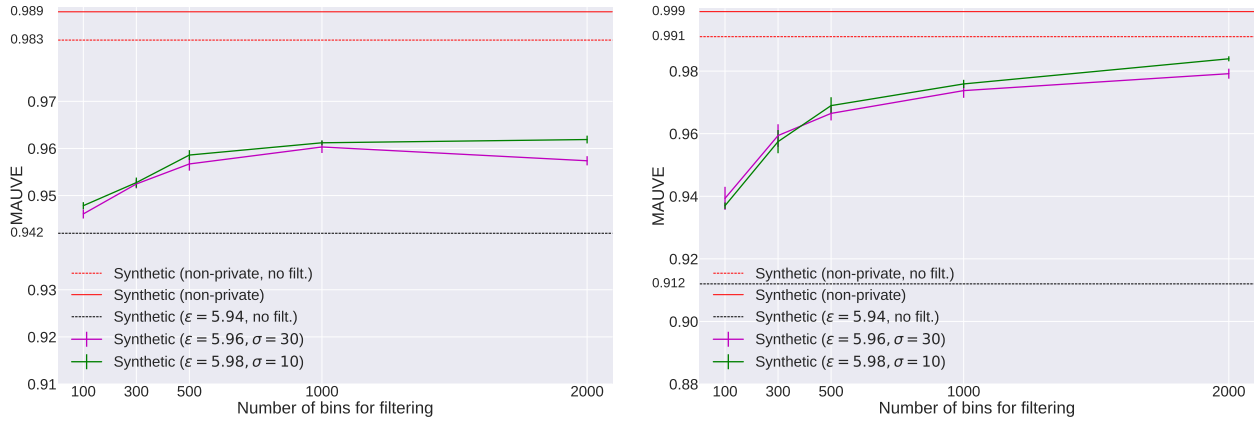
4.2. Measuring the Distributional Gap

We compute the MAUVE score (Pillutla et al., 2021; 2022) between synthetic and real instructions. MAUVE compares two text distributions using divergence frontiers (Kynkäänniemi et al., 2019; Djolonga et al., 2020) and has been widely used to analyze the gap between synthetic text and real text (Yue et al., 2022; Kurakin et al., 2023). We use two representations of text distributions to compute the MAUVE scores: the unigram distribution, and the histogram of clustered text embeddings (Pillutla et al., 2021). Our setup for computing MAUVE scores is in Appendix F.1.

Table 1. MAUVE scores with Chatbot Arena instructions as the target dataset. Best scores of DP models are marked in bold. Filtering synthetic instructions with Algorithm 2 improves the scores of private synthetic instructions by large margins.

	Unigram	Sentence-T5
FLAN (non-private, OOD)	0.191	0.124
Synthetic ($\varepsilon = 2.86$, no filtering)	0.932	0.893
Synthetic ($\varepsilon = 2.91$)	**0.958**	**0.967**
Synthetic ($\varepsilon = 5.94$, no filtering)	0.942	0.912
Synthetic ($\varepsilon = 5.98$)	**0.961**	**0.975**
Synthetic (non-private, no filtering)	0.983	0.991
Synthetic (non-private)	0.989	0.999

Table 1 presents the main MAUVE scores. We compute the scores with three random seeds and report the average.



(a) MAUVE-Unigram

(b) MAUVE-Sentence-T5

Figure 5. Running Algorithm 2 with different K and σ. The MAUVE scores initially improve with an increase in K, then they either start to plateau or decline.

Filtering private synthetic instructions with Algorithm 2 improves MAUVE scores by large margins. We further demonstrate improvements in MAUVE scores for two other datasets, IMDB movie reviews and PubMed abstracts, to underscore the wide applicability of Algorithm 2. Details on the results of the two additional datasets are in Appendix D.4. Additionally, there is a significant distributional gap between FLAN instructions and real-world user instructions. This indicates that in-domain user instructions, while being sensitive, play a crucial role in LLM alignment and should be utilized in a privacy-preserving way.

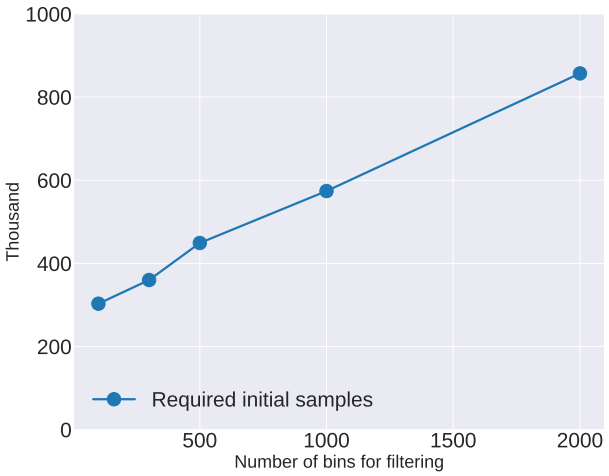


Figure 6. The number of required initial samples for selecting a subset of 180 thousand increases with K.

We also run Algorithm 2 with different values of K and σ. The values of K ranges from 100 to 2000, and the values of σ are chosen from $\{10, 30\}$. The resulting MAUVE scores are presented in Figure 5. The required quantity of initial synthetic instructions as K varies is presented in Figure 6. The MAUVE scores initially improve with an increase in K. However, the scores begin to either plateau or decline for large values of K. This is likely because a larger value of K weakens the aggregated counts, rendering the filtering process more sensitive to noise. Interestingly, the number of initial synthetic instructions required by Algorithm 2 to select a subset of 180 thousand also increases with K. In the rest of this paper, we set $K = 1000$ and $\sigma = 10$ for filtering one million initial synthetic instructions. In Appendix D.3, we run Algorithm 2 with fewer initial synthetic instructions. The results suggest that filtering 500 thousand initial synthetic instructions results in inferior MAUVE scores compared to using one million initial instructions. Notably, to select a subset of 180 thousand from 500 thousand initial samples, we have to use sampling with replacement for large values of K.

4.3. Supervised Fine-tuning

We now assess the utility of synthetic instructions by applying them in supervised instruction fine-tuning. One common way to generate the training data, consisting of pairs of instructions and answers, is to ask human labelers to write the answers (Ouyang et al., 2022; Touvron et al., 2023b). However, human annotation is notably costly both in time and finances. To circumvent this, we follow previous studies (Taori et al., 2023; Chiang et al., 2023) to employ GPT-3.5-Turbo for annotating answers for all instructions, including Chatbot Arena instructions, synthetic instructions, and FLAN instructions. We then fine-tune LLaMA 7B or 13B models on the resulting datasets. All datasets contain 180 thousand pairs of instructions and answers unless otherwise specified. Implementation details are in Appendix F.

We use GPT-4 as a judge to assess the instruction-following capabilities of the fine-tuned models. Specifically, we give GPT-4 one instruction accompanied by two responses generated by the two models being compared, and prompt it to choose a better response. The prompt we used is from Li et al. (2023a). We then compute the average win-rate on a set of evaluation instructions. Recent research indicates that evaluating instruction-following models using GPT-4 is as effective as assessments conducted by humans (Dubois et al., 2023; Zheng et al., 2023c). Furthermore, using GPT-4 for evaluation is scalable, enabling the efficient and cost-effective assessment of dozens of models.

Table 2. Win-rates on Chatbot Arena instructions over 3 seeds. The left column shows the fine-tuning sets (default size is 180K), except for Vicuna-v1.3 which is a public model and is not further fine-tuned. Best win-rates of DP models are in bold. The win rate with filtering ($\varepsilon = 2.91$) surpasses that without filtering ($\varepsilon = 5.94$).

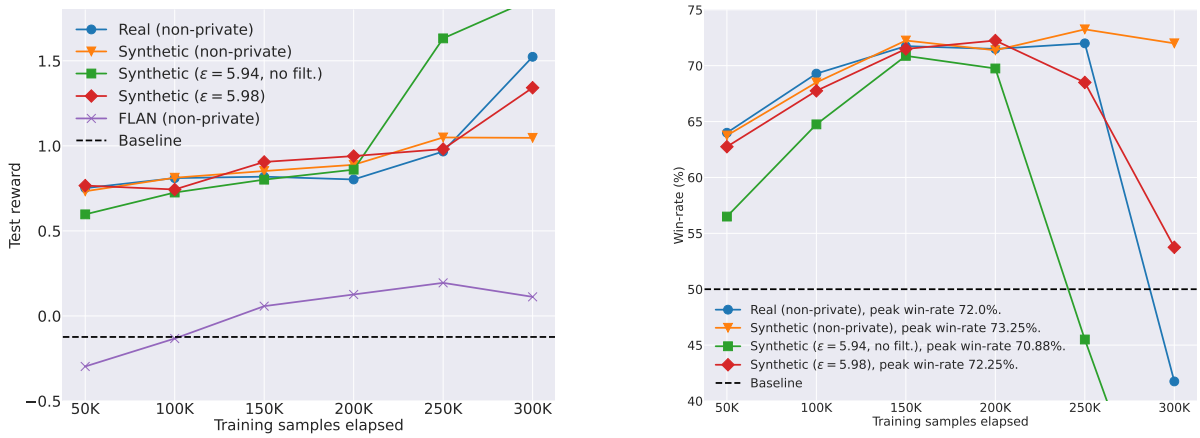
7B Models	
FLAN (non-private)	50%
Vicuna-v1.3	64.1% (±0.61)
Chatbot Arena (non-private)	68.9% (±0.31)
Chatbot Arena ($\varepsilon = 5.94$)	60.7% (±0.42)
Synthetic ($\varepsilon = 5.94$, no filt.)	62.7% (±0.34)
Synthetic ($\varepsilon = 2.91$)	65.6% (±0.36)
Synthetic ($\varepsilon = 5.98$)	67.8% (±0.32)
Synthetic (300K, $\varepsilon = 5.98$)	**68.1%** (±0.37)
13B Models	
Vicuna-v1.3	72.8% (±0.58)
Synthetic ($\varepsilon = 5.94$, no filt.)	71.5% (±0.40)
Synthetic ($\varepsilon = 2.91$)	73.2% (±0.48)
Synthetic ($\varepsilon = 5.98$)	74.0% (±0.62)
Synthetic (300K, $\varepsilon = 5.98$)	**74.5%** (±0.41)

Table 2 presents the win-rates for various models evaluated on 800 random instructions from the Chatbot Arena test split. The baseline model is LLaMA 7B fine-tuned with 180 thousand FLAN instructions. We also run evaluations using the baseline model trained on non-private Chatbot Arena data, with the results presented in Appendix D.5. The findings are consistent with those in Table 2. We include Vicuna-v1.3 (Chiang et al., 2023) in our evaluation, one of the state-of-the-art models that is non-privately fine-tuned from LLaMA on conversations collected from the ShartGPT website (ShareGPT, 2023). The results in Table 2 suggest two main findings. First, private synthetic instructions generated by our algorithms are of high utility, only showing a minor performance drop when compared to using real instructions without any privacy protection. Second, fine-tuning the models on the domain of private instructions yields significantly better performance compared to fine-tuning the models on out-of-domain FLAN instructions. In addition to Chatbot Arena instructions, we also evaluate the models on the AlpacaEval benchmark (Dubois et al., 2023), which comprises 805 instructions collected from public evaluation sets. We present the results in Appendix D.6. It is noteworthy that AlpacaEval instructions do not accurately represent real-world user queries. Nonetheless, such evaluation allows us to make direct comparisons with models listed on the AlpacaEval leaderboard (Li et al., 2023a). When evaluated on AlpacaEval, the best performing DP models slightly underperform Vicuna-v1.3 but are still comparable with other leading opensource models such as LLaMA2-Chat (Touvron et al., 2023b).

In Table 2, we also present the results of private training with real instructions. As the real instructions are not privately released, the corresponding responses are also sensitive. Therefore, we use DP-Adam when training on pairs of real instructions and responses. In contrast, when training with synthetic instructions that have been privately released, we can use standard optimizers because of the post-processing property of DP. It is important to note that even if we use DP-Adam for real instructions, instructions containing personal information are not protected from annotators. As indicated in Table 2, private training with real instructions yields inferior performance compared to training with synthetic instructions. We hypothesize that this is because privatizing responses brings extra cost in model performance. In Figure 18 in Appendix G, we show responses are typically much longer than instructions, making privatizing them a more challenging task.

4.4. RLHF with Proximal Policy Optimization

In this section, we evaluate the utility of our synthetic instructions in Reinforcement Learning from Human Feedback (RLHF). RLHF starts with a baseline model generating initial responses to a set of unlabeled instructions. The responses are then evaluated using some reward models trained to simulate human preference. Then, the baseline model is optimized through RL to increase the reward signal. We use Proximal Policy Optimization (PPO) as the RL algorithm (Schulman et al., 2017; Ouyang et al., 2022). Ideally, the reward model should be developed through private training using (synthetic) real-world user instructions and human-assessed rewards. However, due to the cost of developing reward models, we utilize an off-the-shelf model that comprises 304M parameters and is trained on public human preference datasets. The baseline model is fine-tuned from phi-1.5 (Li et al., 2023b) on the OpenOrca dataset (Lian et al., 2023). Phi-1.5 has 1.3B parameters, which is smaller than the sizes of LLaMA models. We opt for a smaller model because the policy rollout stage in RL is time-consuming for large models. More details regarding the PPO implementation are documented in Appendix F.6.



(a) Average test reward.

(b) Win-rate against the baseline model.

Figure 7. Model performance along the PPO training trajectory. Using synthetic instructions yields similar reward and win-rate curves as using real instructions. In contrast, using non-private but out-of-distribution FLAN instructions yields inferior performance. For real instructions and private synthetic instructions, there is a decline in instruction-following ability in the later stages due to reward over-optimization (Gao et al., 2023).

Our RLHF experiments use four distinct types of data: (1) Chatbot Arena instructions, (2) Synthetic instructions generated by a non-private model, (3) Private synthetic instructions (with/without filtering), and (4) FLAN instructions. We run PPO for 300 thousand instructions. In the cases of synthetic instructions and FLAN instructions, we randomly sample 300 thousand instructions from the corresponding datasets. This means that no instruction is reused during PPO training. In the case of Chatbot Arena instructions, the training makes more than one pass on the training data since its training split contains only 180 thousand samples.

We evaluate checkpoints along the PPO training trajectory, using 800 random instructions from the Chatbot Arena test split. The results are presented in Figure 7. Our evaluation focuses on two key aspects: the average test rewards and win-rates against the baseline model. We use GPT-4-Turbo as the judge and employ the prompt from Li et al. (2023a) to compute win-rates. For both real and private synthetic instructions, we observe an increase in rewards but a decline in instruction-following ability in the later stages of training. This phenomenon is known as reward over-optimization because the reward model serves as an imperfect proxy for actual human preferences, which is an active research topic (Gao et al., 2023). The results, as depicted in Figure 7, demonstrate that the reward and win-rate curves for our synthetic instructions closely mimic those achieved using real instructions. This underscores the high utility of our synthetic instructions.

5. Conclusion

For the common practice of fine-tuning LLMs on instructions, we identify a new threat model on the user provided instructions: potential risks in exposing sensitive personal information in the instructions to human annotators. To address this concern, as well as preventing fine-tuned LLMs to memorize sensitive information in user instructions, we propose to replace user instructions with DP synthetic instructions and introduce a novel framework for generating high-quality instructions by using a DP histogram to resample synthetic instructions to match the distribution of the real dataset in the embedding space. The effectiveness of our framework is validated through comprehensive experiments on publicly available Chatbot Arena datasets (Zheng et al., 2023a;b) and LLaMA and Phi-1.5 models (Touvron et al., 2023a; Li et al., 2023b). Important future directions include developing algorithms capable of privately learning from user instructions in multiple modalities, including image and speech.

Acknowledgement

The authors thank Krishna Pillutla for the insightful discussions on Algorithm 2, and Natalia Ponomareva for her valuable comments and constructive feedback on an early draft.

Impact Statement

In Appendix B, we describe a method for identifying sensitive information within instructions. Our goal is to demonstrate the vulnerability of real-world user inputs, in the hope of encouraging leading AI companies to prioritize the protection of user privacy. While it is possible that potential adversaries might employ the described method, we argue that the negative societal impact of this work is minimal. The method we employ is standard, and the instructions under analysis are already public.

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *ACM SIGSAC conference on computer and communications security*, 2016.

Anil, R., Ghazi, B., Gupta, V., Kumar, R., and Manurangsi, P. Large-scale differentially private bert. *Findings of the Association for Computational Linguistics: EMNLP*, 2022.

Augenstein, S., McMahan, H. B., Ramage, D., Ramaswamy, S., Kairouz, P., Chen, M., Mathews, R., et al. Generative models for effective ml on private, decentralized datasets. *arXiv preprint arXiv:1911.06679*, 2019.

Aydore, S., Brown, W., Kearns, M., Kenthapadi, K., Melis, L., Roth, A., and Siva, A. A. Differentially private query release through adaptive projection. In *International Conference on Machine Learning*. PMLR, 2021.

Azam, S. S., Likhomanenko, T., Pelikan, M., Silovsky, J., et al. Importance of smoothness induced by optimizers in fl4asr: Towards understanding federated learning for end-to-end asr. *arXiv preprint arXiv:2309.13102*, 2023.

Berrada, L., De, S., Shen, J. H., Hayes, J., Stanforth, R., Stutz, D., Kohli, P., Smith, S. L., and Balle, B. Unlocking accuracy and fairness in differentially private image classification. *arXiv preprint arXiv:2308.10888*, 2023.

Bommasani, R., Wu, S., and Schofield, X. Towards private synthetic text generation. In *NeurIPS 2019 Machine Learning with Guarantees Workshop*, 2019.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 2020.

Bu, Z., Wang, Y.-X., Zha, S., and Karypis, G. Differentially private bias-term only fine-tuning of foundation models. *arXiv preprint arXiv:2210.00036*, 2022.

Bu, Z., Wang, Y.-X., Zha, S., and Karypis, G. Differentially private optimization on large model at small cost. In *International Conference on Machine Learning*. PMLR, 2023.

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S., et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

Carlini, N., Liu, C., Erlingsson, Ú., Kos, J., and Song, D. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *USENIX Security Symposium*, 2019.

Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U., et al. Extracting training data from large language models. *USENIX Security Symposium*, 2021.

Carlini, N., Hayes, J., Nasr, M., Jagielski, M., Sehwag, V., Tramer, F., Balle, B., Ippolito, D., and Wallace, E. Extracting training data from diffusion models. *arXiv preprint arXiv:2301.13188*, 2023.

Chiang, W.-L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J. E., Stoica, I., and Xing, E. P. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL https://lmsys.org/blog/2023-03-30-vicuna/.

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022.

De, S., Berrada, L., Hayes, J., Smith, S. L., and Balle, B. Unlocking high-accuracy differentially private image classification through scale. *arXiv preprint arXiv:2204.13650*, 2022.

Ding, Y. and Wu, X. Revisiting hyperparameter tuning with differential privacy. *arXiv preprint arXiv:2211.01852*, 2022.

Djolonga, J., Lucic, M., Cuturi, M., Bachem, O., Bousquet, O., and Gelly, S. Precision-recall curves using information divergence frontiers. In *International Conference on Artificial Intelligence and Statistics*, 2020.

Dockhorn, T., Cao, T., Vahdat, A., and Kreis, K. Differentially private diffusion models. *arXiv preprint arXiv:2210.09929*, 2022.

Du, M., Yue, X., Chow, S. S., Wang, T., Huang, C., and Sun, H. Dp-forward: Fine-tuning and inference on language models with differential privacy in forward pass. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2665–2679, 2023.

Duan, H., Dziedzic, A., Papernot, N., and Boenisch, F. Flocks of stochastic parrots: Differentially private prompt learning for large language models. *arXiv preprint arXiv:2305.15594*, 2023a.

Duan, H., Dziedzic, A., Yaghini, M., Papernot, N., and Boenisch, F. On the privacy risk of in-context learning. In *The 61st Annual Meeting Of The Association For Computational Linguistics*, 2023b.

Duan, H., Dziedzic, A., Papernot, N., and Boenisch, F. Flocks of stochastic parrots: Differentially private prompt learning for large language models. *Advances in Neural Information Processing Systems*, 36, 2024.

Dubois, Y., Li, X., Taori, R., Zhang, T., Gulrajani, I., Ba, J., Guestrin, C., Liang, P., and Hashimoto, T. B. Alpacafarm: A simulation framework for methods that learn from human feedback. *arXiv preprint arXiv:2305.14387*, 2023.

Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., and Naor, M. Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2006a.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, 2006b.

Dwork, C., Roth, A., et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 2014.

Ganesh, A., Haghifam, M., Nasr, M., Oh, S., Steinke, T., Thakkar, O., Thakurta, A. G., and Wang, L. Why is public pretraining necessary for private model training? In *International Conference on Machine Learning*, 2023.

Gao, L., Schulman, J., and Hilton, J. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, 2023.

Ge, C., Mohapatra, S., He, X., and Ilyas, I. F. Kamino: Constraint-aware differentially private data synthesis. *arXiv preprint arXiv:2012.15713*, 2020.

Geng, X., Gudibande, A., Liu, H., Wallace, E., Abbeel, P., Levine, S., and Song, D. Koala: A dialogue model for academic research. *Blog post, April*, 2023.

Ghalebikesabi, S., Berrada, L., Gowal, S., Ktena, I., Stanforth, R., Hayes, J., De, S., Smith, S. L., Wiles, O., and Balle, B. Differentially private diffusion models generate useful synthetic images. *arXiv preprint arXiv:2302.13861*, 2023.

Ghazi, B., Kamath, P., Kumar, R., and Manurangsi, P. Faster privacy accounting via evolving discretization. In *International Conference on Machine Learning*, 2022.

Golatkar, A., Achille, A., Wang, Y.-X., Roth, A., Kearns, M., and Soatto, S. Mixed differential privacy in computer vision. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Gopi, S., Lee, Y. T., and Wutschitz, L. Numerical composition of differential privacy. *Advances in Neural Information Processing Systems*, 2021.

Guo, C., Karrer, B., Chaudhuri, K., and van der Maaten, L. Bounding training data reconstruction in private (deep) learning. In *International Conference on Machine Learning*, 2022.

Harder, F., Adamczewski, K., and Park, M. Dp-merf: Differentially private mean embeddings with randomfeatures for practical privacy-preserving data generation. In *International conference on artificial intelligence and statistics*. PMLR, 2021.

He, J., Li, X., Yu, D., Zhang, H., Kulkarni, J., Lee, Y. T., Backurs, A., Yu, N., and Bian, J. Exploring the limits of differentially private deep learning with group-wise clipping. *International Conference on Learning Representations*, 2023.

Holtzman, A., Buys, J., Du, L., Forbes, M., and Choi, Y. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.

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

Hong, J., Wang, J. T., Zhang, C., Li, Z., Li, B., and Wang, Z. Dp-opt: Make large language model your privacy-preserving prompt engineer. *International Conference on Learning Representations*, 2024.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pp. 2790–2799. PMLR, 2019.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation

of large language models. *International Conference on Learning Representations*, 2022.

Jagielski, M. A note on interpreting canary exposure. *arXiv preprint arXiv:2306.00133*, 2023.

Johnson, J., Douze, M., and Jégou, H. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 2019.

Jordon, J., Yoon, J., and Van Der Schaar, M. Pate-gan: Generating synthetic data with differential privacy guarantees. In *International conference on learning representations*, 2018.

Kairouz, P., Oh, S., and Viswanath, P. The composition theorem for differential privacy. In *International conference on machine learning*, 2015.

Kairouz, P., McMahan, B., Song, S., Thakkar, O., Thakurta, A., and Xu, Z. Practical and private (deep) learning without sampling or shuffling. In *International Conference on Machine Learning*, pp. 5213–5225. PMLR, 2021.

Kandpal, N., Wallace, E., and Raffel, C. Deduplicating training data mitigates privacy risks in language models. In *International Conference on Machine Learning*. PMLR, 2022.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015.

Koskela, A., Jälkö, J., and Honkela, A. Computing tight differential privacy guarantees using fft. In *International Conference on Artificial Intelligence and Statistics*, 2020.

Kurakin, A., Song, S., Chien, S., Geambasu, R., Terzis, A., and Thakurta, A. Toward training at imagenet scale with differential privacy. *arXiv preprint arXiv:2201.12328*, 2022.

Kurakin, A., Ponomareva, N., Syed, U., MacDermed, L., and Terzis, A. Harnessing large-language models to generate private synthetic text. *arXiv preprint arXiv:2306.01684*, 2023.

Kynkäänniemi, T., Karras, T., Laine, S., Lehtinen, J., and Aila, T. Improved precision and recall metric for assessing generative models. *Advances in Neural Information Processing Systems*, 32, 2019.

Lee, K., Ippolito, D., Nystrom, A., Zhang, C., Eck, D., Callison-Burch, C., and Carlini, N. Deduplicating training data makes language models better. *arXiv preprint arXiv:2107.06499*, 2021.

Li, X., Liu, D., Hashimoto, T. B., Inan, H. A., Kulkarni, J., Lee, Y.-T., and Guha Thakurta, A. When does differentially private learning not suffer in high dimensions? *Advances in Neural Information Processing Systems*, 2022a.

Li, X., Tramer, F., Liang, P., and Hashimoto, T. Large language models can be strong differentially private learners. *International Conference on Learning Representations*, 2022b.

Li, X., Zhang, T., Dubois, Y., Taori, R., Gulrajani, I., Guestrin, C., Liang, P., and Hashimoto, T. B. Alpacaeval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval, 2023a.

Li, Y., Bubeck, S., Eldan, R., Del Giorno, A., Gunasekar, S., and Lee, Y. T. Textbooks are all you need ii: phi-1.5 technical report. *arXiv preprint arXiv:2309.05463*, 2023b.

Lian, W., Goodson, B., Pentland, E., Cook, A., Vong, C., and "Teknium". Openorca: An open dataset of gpt augmented flan reasoning traces. https://https://huggingface.co/Open-Orca/OpenOrca, 2023.

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

Liu, J. and Talwar, K. Private selection from private candidates. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, 2019.

Liu, T., Vietri, G., and Wu, S. Z. Iterative methods for private synthetic data: Unifying framework and new methods. *Advances in Neural Information Processing Systems*, 2021.

Liu, T., Tang, J., Vietri, G., and Wu, S. Generating private synthetic data with genetic algorithms. In *International Conference on Machine Learning*, pp. 22009–22027. PMLR, 2023.

Longpre, S., Hou, L., Vu, T., Webson, A., Chung, H. W., Tay, Y., Zhou, D., Le, Q. V., Zoph, B., Wei, J., et al. The flan collection: Designing data and methods for effective instruction tuning. In *International Conference on Machine Learning*, pp. 22631–22648. PMLR, 2023.

Lowy, A., Li, Z., Liu, J., Koike-Akino, T., Parsons, K., and Wang, Y. Why does differential privacy with large epsilon defend against practical membership inference attacks? *arXiv preprint arXiv:2402.09540*, 2024.

Luo, Z., Wu, D. J., Adeli, E., and Fei-Fei, L. Scalable differential privacy with sparse network finetuning. In

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021.

Maas, A., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pp. 142–150, 2011.

Majmudar, J., Dupuy, C., Peris, C., Smaili, S., Gupta, R., and Zemel, R. Differentially private decoding in large language models. *arXiv preprint arXiv:2205.13621*, 2022.

Mattern, J., Jin, Z., Weggenmann, B., Schoelkopf, B., and Sachan, M. Differentially private language models for secure data sharing. *arXiv preprint arXiv:2210.13918*, 2022.

McKenna, R., Miklau, G., and Sheldon, D. Winning the nist contest: A scalable and general approach to differentially private synthetic data. *arXiv preprint arXiv:2108.04978*, 2021.

McKenna, R., Mullins, B., Sheldon, D., and Miklau, G. Aim: An adaptive and iterative mechanism for differentially private synthetic data. *arXiv preprint arXiv:2201.12677*, 2022.

Mehta, H., Thakurta, A., Kurakin, A., and Cutkosky, A. Large scale transfer learning for differentially private image classification. *arXiv preprint arXiv:2205.02973*, 2022.

Mohapatra, S., Sasy, S., He, X., Kamath, G., and Thakkar, O. The role of adaptive optimizers for honest private hyperparameter selection. In *Proceedings of the AAAI conference on artificial intelligence*, 2022.

Narayanan, A. and Shmatikov, V. Robust de-anonymization of large sparse datasets. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pp. 111–125. IEEE, 2008.

Nasr, M., Carlini, N., Hayase, J., Jagielski, M., Cooper, A. F., Ippolito, D., Choquette-Choo, C. A., Wallace, E., Tramèr, F., and Lee, K. Scalable extraction of training data from (production) language models. *arXiv preprint arXiv:2311.17035*, 2023a.

Nasr, M., Hayes, J., Steinke, T., Balle, B., Tramèr, F., Jagielski, M., Carlini, N., and Terzis, A. Tight auditing of differentially private machine learning. *arXiv preprint arXiv:2302.07956*, 2023b.

Neunhoeffer, M., Wu, Z. S., and Dwork, C. Private post-gan boosting. *arXiv preprint arXiv:2007.11934*, 2020.

Ni, J., Ábrego, G. H., Constant, N., Ma, J., Hall, K. B., Cer, D., and Yang, Y. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. *arXiv preprint arXiv:2108.08877*, 2021.

OpenAI. Aligning language models to follow instructions, 2022. URL https://openai.com/research/instruction-following. Accessed: 2023-12-31.

OpenAI. Openai privacy policy, 2023. URL https://openai.com/policies/privacy-policy. Accessed: 2023-12-31.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 2022.

Panda, A., Tang, X., Sehwag, V., Mahloujifar, S., and Mittal, P. Dp-raft: A differentially private recipe for accelerated fine-tuning. *arXiv preprint arXiv:2212.04486*, 2022.

Panda, A., Wu, T., Wang, J. T., and Mittal, P. Differentially private in-context learning. *arXiv preprint arXiv:2305.01639*, 2023.

Papernot, N. and Steinke, T. Hyperparameter tuning with renyi differential privacy. *International Conference on Learning Representations*, 2022.

Pilán, I., Lison, P., Øvrelid, L., Papadopoulou, A., Sánchez, D., and Batet, M. The text anonymization benchmark (tab): A dedicated corpus and evaluation framework for text anonymization. *Computational Linguistics*, 48(4): 1053–1101, 2022.

Pillutla, K., Swayamdipta, S., Zellers, R., Thickstun, J., Welleck, S., Choi, Y., and Harchaoui, Z. Mauve: Measuring the gap between neural text and human text using divergence frontiers. *Advances in Neural Information Processing Systems*, 34:4816–4828, 2021.

Pillutla, K., Liu, L., Thickstun, J., Welleck, S., Swayamdipta, S., Zellers, R., Oh, S., Choi, Y., and Harchaoui, Z. Mauve scores for generative models: Theory and practice. *arXiv preprint arXiv:2212.14578*, 2022.

Pillutla, K., Andrew, G., Kairouz, P., McMahan, H. B., Oprea, A., and Oh, S. Unleashing the power of randomization in auditing differentially private ml. *arXiv preprint arXiv:2305.18447*, 2023.

Ponomareva, N., Hazimeh, H., Kurakin, A., Xu, Z., Denison, C., McMahan, H. B., Vassilvitskii, S., Chien, S., and Thakurta, A. G. How to dp-fy ml: A practical guide to machine learning with differential privacy. *Journal of Artificial Intelligence Research*, 77:1113–1201, 2023.

Putta, P., Steele, A., and Ferrara, J. W. Differentially private conditional text generation for synthetic data production. 2022.

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

Rahman, M. A., Rahman, T., Laganiere, R., Mohammed, N., and Wang, Y. Membership inference attack against differentially private deep learning model. *Transactions on Data Privacy*, 2018.

Redberg, R. E., Koskela, A., and Wang, Y.-X. Improving the privacy and practicality of objective perturbation for differentially private linear learners. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Rosenblatt, L., Liu, X., Pouyanfar, S., de Leon, E., Desai, A., and Allen, J. Differentially private synthetic data: Applied evaluations and enhancements. *arXiv preprint arXiv:2011.05537*, 2020.

Sander, T., Stock, P., and Sablayrolles, A. Tan without a burn: Scaling laws of dp-sgd. *arXiv preprint arXiv:2210.03403*, 2022.

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

Sehwag, V., Panda, A., Pokle, A., Tang, X., Mahloujifar, S., Chiang, M., Kolter, J. Z., and Mittal, P. Differentially private generation of high fidelity samples from diffusion models. 2023.

Shamsabadi, A. S., Srivastava, B. M. L., Bellet, A., Vauquier, N., Vincent, E., Maouche, M., Tommasi, M., and Papernot, N. Differentially private speaker anonymization. *arXiv preprint arXiv:2202.11823*, 2022.

ShareGPT. Sharegpt:share your wildest chatgpt conversations with one click., 2023. URL https://sharegpt.com/. Accessed: 2023-12-31.

Song, S., Chaudhuri, K., and Sarwate, A. D. Stochastic gradient descent with differentially private updates. In *Global Conference on Signal and Information Processing*, 2013.

Steinke, T. Composition of differential privacy & privacy amplification by subsampling. *arXiv preprint arXiv:2210.00597*, 2022.

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

Stock, P., Shilov, I., Mironov, I., and Sablayrolles, A. Defending against reconstruction attacks with r\'enyi differential privacy. *arXiv preprint arXiv:2202.07623*, 2022.

Tan, R. and Cabato, R. Behind the ai boom, an army of overseas workers in 'digital sweatshops'. In *The Washington Post*, 2023.

Tang, X., Panda, A., Sehwag, V., and Mittal, P. Differentially private image classification by learning priors from random processes. *arXiv preprint arXiv:2306.06076*, 2023a.

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

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

Tang, X., Shin, R., Inan, H. A., Manoel, A., Mireshghallah, F., Lin, Z., Gopi, S., Kulkarni, J., and Sim, R. Privacy-preserving in-context learning with differentially private few-shot generation. *International Conference on Learning Representations*, 2024b.

Tantipongpipat, U., Waites, C., Boob, D., Siva, A. A., and Cummings, R. Differentially private mixed-type data generation for unsupervised learning. corr abs/1912.03250 (2019). *arXiv preprint arXiv:1912.03250*, 2019.

Tao, Y., McKenna, R., Hay, M., Machanavajjhala, A., and Miklau, G. Benchmarking differentially private synthetic data generation algorithms. *arXiv preprint arXiv:2112.09238*, 2021.

Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., and Hashimoto, T. B. 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, 2023.

Thaker, P., Setlur, A., Wu, Z. S., and Smith, V. Leveraging public representations for private transfer learning. *arXiv preprint arXiv:2312.15551*, 2023.

Tobaben, M., Shysheya, A., Bronskill, J., Paverd, A., Tople, S., Zanella-Beguelin, S., Turner, R. E., and Honkela, A. On the efficacy of differentially private few-shot image classification. *Transactions on Machine Learning Research*, 2023.

Torkzadehmahani, R., Kairouz, P., and Paten, B. Dp-cgan: Differentially private synthetic data and label generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

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

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

Tramèr, F., Kamath, G., and Carlini, N. Considerations for differentially private learning with large-scale public pretraining. *arXiv preprint arXiv:2212.06470*, 2022.

Vadhan, S. The complexity of differential privacy. *Tutorials on the Foundations of Cryptography: Dedicated to Oded Goldreich*, pp. 347–450, 2017.

Vietri, G., Tian, G., Bun, M., Steinke, T., and Wu, S. New oracle-efficient algorithms for private synthetic data release. In *International Conference on Machine Learning*, pp. 9765–9774. PMLR, 2020.

Vietri, G., Archambeau, C., Aydore, S., Brown, W., Kearns, M., Roth, A., Siva, A., Tang, S., and Wu, S. Z. Private synthetic data for multitask learning and marginal queries. *Advances in Neural Information Processing Systems*, 2022.

Völske, M., Potthast, M., Syed, S., and Stein, B. Tl; dr: Mining reddit to learn automatic summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, pp. 59–63, 2017.

Wang, B., Zhang, Y. J., Cao, Y., Li, B., McMahan, H. B., Oh, S., Xu, Z., and Zaheer, M. Can public large language models help private cross-device federated learning? *arXiv preprint arXiv:2305.12132*, 2023a.

Wang, H., Sudalairaj, S., Henning, J., Greenewald, K., and Srivastava, A. Post-processing private synthetic data for improving utility on selected measures. *Advances in Neural Information Processing Systems*, 2023b.

Wang, Y., Kordi, Y., Mishra, S., Liu, A., Smith, N. A., Khashabi, D., and Hajishirzi, H. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560*, 2022.

Wei, J., Bosma, M., Zhao, V. Y., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., and Le, Q. V. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

Wu, F., Inan, H. A., Backurs, A., Chandrasekaran, V., Kulkarni, J., and Sim, R. Privately aligning language models with reinforcement learning. *International Conference on Learning Representations*, 2024a.

Wu, T., Panda, A., Wang, J. T., and Mittal, P. Privacy-preserving in-context learning for large language models. In *International Conference on Learning Representations*, 2024b.

Xiao, Y., Jin, Y., Bai, Y., Wu, Y., Yang, X., Luo, X., Yu, W., Zhao, X., Liu, Y., Chen, H., et al. Large language models can be good privacy protection learners. *arXiv preprint arXiv:2310.02469*, 2023.

Xie, L., Lin, K., Wang, S., Wang, F., and Zhou, J. Differentially private generative adversarial network. *arXiv preprint arXiv:1802.06739*, 2018.

Xu, Z., Collins, M., Wang, Y., Panait, L., Oh, S., Augenstein, S., Liu, T., Schroff, F., and McMahan, H. B. Learning to generate image embeddings with user-level differential privacy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7969–7980, 2023.

Yu, D., Zhang, H., Chen, W., Yin, J., and Liu, T.-Y. Large scale private learning via low-rank reparametrization. In *International Conference on Machine Learning*, 2021.

Yu, D., Naik, S., Backurs, A., Gopi, S., Inan, H. A., Kamath, G., Kulkarni, J., Lee, Y. T., Manoel, A., Wutschitz, L., Yekhanin, S., and Huishuai, Z. Differentially private fine-tuning of language models. *International Conference on Learning Representations*, 2022.

Yue, X., Inan, H. A., Li, X., Kumar, G., McAnallen, J., Sun, H., Levitan, D., and Sim, R. Synthetic text generation with differential privacy: A simple and practical recipe. *arXiv preprint arXiv:2210.14348*, 2022.

Zhang, X., Bu, Z., Wu, Z. S., and Hong, M. Differentially private sgd without clipping bias: An error-feedback approach. *arXiv preprint arXiv:2311.14632*, 2023a.

Zhang, Z., Jia, M., Yao, B., Das, S., Lerner, A., Wang, D., and Li, T. It's a fair game, or is it? examining how users navigate disclosure risks and benefits when using llm-based conversational agents. *arXiv preprint arXiv:2309.11653*, 2023b.

Zhao, W., Ren, X., Hessel, J., Cardie, C., Choi, Y., and Deng, Y. (inthe)wildchat: 570k chatGPT interaction logs in the wild. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Bl8u7ZRlbM.

Zheng, L., Chiang, W.-L., Sheng, Y., Li, T., Zhuang, S., Wu, Z., Zhuang, Y., Li, Z., Lin, Z., Xing, E., et al. Lmsys-chat-1m: A large-scale real-world llm conversation dataset. *arXiv preprint arXiv:2309.11998*, 2023a.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., et al. The chatbot arena conversations dataset., 2023b. URL https://huggingface.co/datasets/lmsys/chatbot_arena_conversations. Accessed: 2023-12-31.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*, 2023c.

Zhu, Y., Zhao, X., Guo, C., and Wang, Y.-X. Private prediction strikes back! private kernelized nearest neighbors with individual renyi filter. *arXiv preprint arXiv:2306.07381*, 2023.

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

A. Additional Related Work

Wu et al. (2024a) present pioneering efforts in aligning language models with differential privacy. They align GPT-2 models (Radford et al., 2019) using private optimizers to ensure that models trained with both supervised fine-tuning and reinforcement learning satisfy differential privacy. They demonstrate that the rewards achieved by DP models are comparable to those of non-private models on the IMDb dataset (Maas et al., 2011) and the Reddit TL;DR dataset (Völske et al., 2017). Compared to Wu et al. (2024a), this work introduces several new aspects. First, our framework mitigates privacy risks arising from both human annotation and model memorization. In contrast, the privacy guarantees in Wu et al. (2024a) only apply to the trained models. Second, our experiments focus on instruction datasets collected during the real-world deployment of LLMs (Zheng et al., 2023a). Moreover, we evaluate the aligned models' ability to follow instructions not only through rewards but also through win-rates evaluated by LLM judges (Zheng et al., 2023c; Dubois et al., 2023).

Lin et al. (2023) investigate generating realistic differentially private synthetic images using only API access to foundation models. Their approach, termed Private Evolution, also leverages votes from real samples to select synthetic samples. Private Evolution is a multi-round generation algorithm. In each round, it first generates a pool of candidate images by using variation APIs to create variations for samples from the previous round. Then they sample (with replacement) from the candidate pool based on votes from real images. The key difference between their selection process with ours is that we cluster synthetic samples to improve the signal-to-noise ratio during the selection process. Our Algorithm 2 allows us to run the selection process with a minimal privacy cost. Additionally, we focus on training instruction-following models and conduct comprehensive experiments to verify the effectiveness of our algorithms.

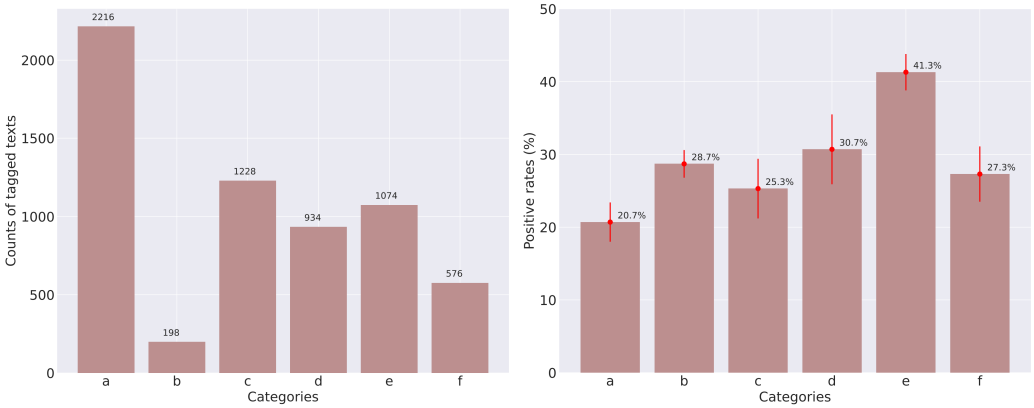
Recently, Zhang et al. (2023b) show that users may disclose private information in their conversations with LLMs. They investigate the conversations collected on the ShareGPT website. In this work, we investigate the user instructions in the Chatbot Arena Conversations dataset[1]. Compared to the analysis in Zhang et al. (2023b), our analysis is new in two aspects. First, all conversations from ShareGPT are proactively shared by the user themselves and hence may result in an inherent sample bias. In contrast, users instructions from Chatbot Arena are directly collected by the service provider. Second, we conduct a quantitative analysis of six distinct categories of personally identifiable information, providing a more comprehensive understanding for the presence of common types of sensitive information.

In addition to DP synthetic text, significant research has been dedicated to generating other forms of data with differential privacy, including synthetic tabular data (Jordon et al., 2018; Tantipongpipat et al., 2019; Ge et al., 2020; Tao et al., 2021; Rosenblatt et al., 2020; Aydore et al., 2021; Vietri et al., 2020; 2022; McKenna et al., 2021; 2022; Liu et al., 2023) and synthetic image data (Xie et al., 2018; Torkzadehmahani et al., 2019; Harder et al., 2021; Dockhorn et al., 2022; Ghalebikesabi et al., 2023; Sehwag et al., 2023). In the realm of generating DP synthetic tabular data, recent studies have explored enhancing the quality of synthetic data through post-processing the initial outputs (Neunhoeffer et al., 2020; Liu et al., 2021; Wang et al., 2023b). Most recently, Wang et al. (2023b) propose to privatize the correlation matrix of a private dataset, and then re-sample the initial synthetic data based on the privatized correlation matrix. Their approach achieves state-of-the-art performance on several benchmarks. To the best of our knowledge, this work marks the first attempt to investigate the post-processing of private synthetic text. We introduce a filtering algorithm that demonstrates both high effectiveness and minimal additional privacy costs.

Leveraging pre-trained models in private learning has substantially improved the privacy-utility trade-off of deep learning with DP. This advancement extends to various domains, including natural language processing (Majmudar et al., 2022; Bu et al., 2023; Wang et al., 2023a; Du et al., 2023; Xiao et al., 2023; Duan et al., 2023a; Tang et al., 2023b; 2024a; Panda et al., 2023; Hong et al., 2023), computer vision (Luo et al., 2021; Golatkar et al., 2022; Mehta et al., 2022; Panda et al., 2022; De et al., 2022; Zhu et al., 2023; Tang et al., 2023a; Xu et al., 2023; Thaker et al., 2023; Zhang et al., 2023a; Tobaben et al., 2023), and speech recognition (Shamsabadi et al., 2022; Azam et al., 2023). Recent theoretical studies have provided evidence to explain the efficacy of using pre-trained models in private learning (Li et al., 2022a; Ganesh et al., 2023). In this paper, we follow this line of research and utilize DP fine-tuning to train generative models for creating synthetic instructions. It is noteworthy that if the pre-training process does not satisfy differential privacy, then DP fine-tuning only assures the privacy of fine-tuning data. In real-world deployments, it is important to also consider the privacy risks of pre-training data (Tramèr et al., 2022). One potential solution is to also enforce differential privacy in the pre-training stage (Kurakin et al., 2022; Anil et al., 2022; Sander et al., 2022).

[1] https://huggingface.co/datasets/lmsys/chatbot_arena_conversations

B. Real-world User Instructions Are Sensitive



(a) Counts of PII by category in the initial screening.

(b) True positive rates in human annotation.

Figure 8. Main results of our PII analysis. The x-axis represents six categories of sensitive text pieces: a. full names of real individuals, b. unique identifiers such as email addresses, c. specific locations, d. names of organisations, e. occupational titles, and f. specific times. Figure (a) shows the results of initial screening by prompting GPT-3.5-Turbo to flag sensitive text pieces in user instructions. Figure (b) presents the true positive rates of these initial screening results, as determined by human annotation.

We conduct a systematic analysis to demonstrate the prevalence of private information in real-world user instructions. We investigate around 12 thousand instructions from the Chatbot Arena Conversations dataset (Zheng et al., 2023b). The dataset has 33 thousand conversations with human preferences collected on the Chatbot Arena website, and each conversation sample contains a user instruction, two responses from two different models, and preference from the user. In our analysis, we only use the user instructions, and concatenate the instructions of different rounds into one string for multi-round conversations. We deduplicate the instructions and only keep instructions that have more than 20 GPT-3.5-Turbo tokens, which reduces the number of instructions from 33 thousand to 12,472 instructions.

```
1. For the given instruction from a user (denoted by INSTRUCTION), identify the following personally
identifiable information.
a. The full names of real individuals.
b. Number or code that identifies entities in real word, such as phone number, email address, personal or
organization website.
c. Specific places and locations, such as: cities, areas, named infrastructures (bus stops, bridges, etc.).
d. Names of organisations, such as companies, schools, universities.
e. Occupational titles, such as job names, position names.
f. Description of a specific time, such as October 3, 2018 or 13th June.

2. List the answer in the following format: [[catergory]]: personally identifiable information.
3. If a personally identifiable information is mentioned multiple times, only count it once.
4. Do not include personally identifiable information in public articles (such as news) or fiction stories.


DEMONSTRATIONS: ...

INSTRUCTION = ...
RESULT:
```

Figure 9. The prompt for initial screening with GPT-3.5-Turbo. The categories of sensitive texts are inspired by the categories of personally identifiable information in Pilán et al. (2022). The authors label two emails from the Enron Email dataset as demonstrations.

Inspired by the personally identifiable information (PII) categories in Pilán et al. (2022), we consider six categories of sensitive text pieces, including: a. full names of real individuals, b. unique identifiers such as email addresses, c. specific locations, d. names of organisations, e. occupational titles, and f. specific times. We first prompt GPT-3.5-Turbo to flag text pieces that might be sensitive as the first round of screening. We choose LLM-based PII detection instead of rule-based detection because recent studies suggest LLMs achieve a higher detection rate for PII than existing rule-based detection tools (Bubeck et al., 2023). We show the prompt used for LLM-based PII detection in Figure 9 and the counts of flagged

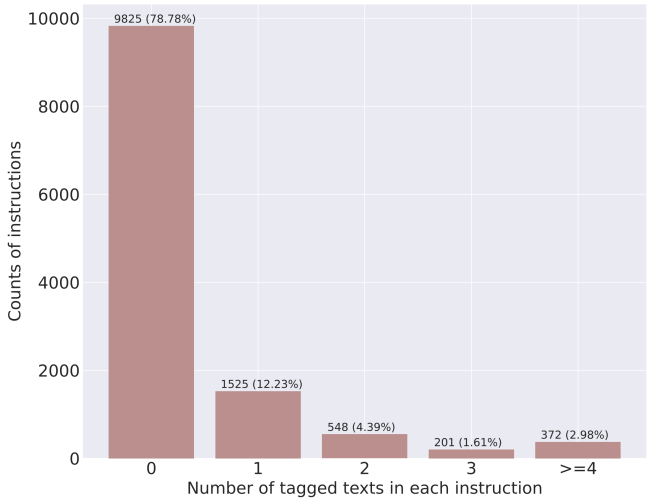


Figure 10. The number of instructions containing varying numbers of text pieces that are flagged by GPT-3.5-Turbo as sensitive.

text pieces by category in Figure 8a. Out of the 12,472 instructions being analyzed, more than 6000 text pieces in total, ranging in length from a single word to a few words, are flagged by GPT-3.5, and more than 20% of the instructions contain at least one piece of sensitive text. The histogram of the number of instructions containing varying numbers of flagged text pieces is presented in Figure 10.

```
Annotation guide: Below is a piece of information tagged as sensitive by GPT-3.5, along with its
accompanying instruction. Does it appear to relate to real individuals and should be considered as
sensitive? Do NOT try to identify the user in the real world.

Tagged text: business CEOs
Instruction: are creative jobs, business CEOs, and social jobs safe from being replaced by AGI? explain
thoroughly.
Label: not sensitive.

Tagged text: Marketing Director
Instruction: Please give me a profile summary based on my work experience: currently [Job title and company
name] in [Location]. Previously 19 years in [Company name], promoted from [Job title] to [Job title] and
then to Marketing Director for ......
Label: sensitive.
```

Figure 11. Annotation guideline and sample annotations made by the authors. Sensitive text pieces, expected for the one that is being annotated, are redacted by the authors to protect user privacy.

While the initial screening by LLM marks many text pieces as sensitive, we find that many of the flagged text pieces are false positives that are not concerning. For instance, names in fictional stories or public news are often identified as sensitive by GPT-3.5, even if we instruct the LLM to not flag them. We manually check the flagged text pieces to estimate the true positive rates. We randomly sample 50 flagged pieces from each of the six categories. And for each tagged text, the annotators label whether it is sensitive (true positive) or not. The annotation guideline and two sample annotations are shown in Figure 11. We repeat this process for three times with different random seeds for sampling and 150 flagged text pieces are manually reviewed for each category. The average positive rates and standard deviations are in Figure 8b. The average positive rate is larger than 20% for every category. The category with the lowest and highest positive rates are 'full names of real individuals' (20.7%) and 'occupational titles' (41.3%), respectively. In Figure 12, we present common types of false positive predictions made by GPT-3.5. Examples of sensitive instructions found are shown in Figure 2.

Type: ChatGPT makes false predictions.
Sample instruction: Which of the following is NOT a component of the 1995 Dodge Viper mechanical system? A: Toaster oven, B: Flywheel, C: Clutch, D: Exhaust headers

Type: The entities are either public figures or fictional characters.
Sample instruction: Write a compare and contrast essay about Tom Buchanan and Jay Gatsby from F. Scott Fitzgerald's The Great Gatsby.

Type: The entities are from a public news.
Sample instruction: translate to spanish
""" (Reuters) - International Business Machines Corp expects to pause hiring for roles as roughly 7,800 jobs ....., CEO Arvind Krishna told Bloomberg News on Monday. """

Figure 12. Common types of false positive sensitive pieces that are flagged by GPT-3.5 but then annotated as not sensitive by the authors.

C. Empirical Privacy Leakage

Repetition: 1
Canary: Make the ending of this email more friendly.
Thanks in advance for your help! Looking forward to hearing from you.
Thanks,
Jay
Tel: 716-802-3549
Canary: Convert this address into json format 'street':, 'city':,'state':.
172 Homestead Street, Manchester, CT.

Repetition: 10
Canary: Which state does this phone number belong to?
Number: (253) 697-4180
Canary: Create a travel itinerary to Canada. Include one night stay at my friend's place: 531 Bobbin Mill Road, Lunenburg, VT.

Repetition: 100
Canary: Does this text message sound persuasive?
Please call us at 643 827 1509 ASAP otherwise you will lose your discount.
Canary: I didn't receive my package, what should I reply to this email?
Your package to 2715 Thornbrook Court, Odenton, MD has been delivered. Please contact us if you need assistance.

Figure 13. Six canary instructions. Each canary contains either a fake phone number or a fake address.

One common empirical approach to assess the privacy risk of a training algorithm is injecting canaries (specifically crafted training samples) into the training set, and evaluating the extent to which the canaries are exposed through querying the trained model (Pillutla et al., 2023; Nasr et al., 2023b; Jagielski, 2023). In this section, we follow this approach to compare the empirical privacy leakage of models trained with and without DP. We use six canary instructions, three of them contain phone numbers and the other three contain addresses (see Figure 13 for the six canaries). Phone numbers are randomized 10-digit sequences, and addresses are generated using an off-the-shelf tool[2]. Following Yue et al. (2022), we repeat the canaries with different times (1, 10, and 100) to explore the worst case privacy leakage (Lee et al., 2021; Kandpal et al., 2022). It is important to note that the privacy parameter ε for a canary linearly increases with the number of repetitions (Dwork et al., 2014; Vadhan, 2017).

We quantify the leakage using two measures. The first measure is the relative position of a canary's loss among the losses of 10,000 instructions. These instructions follow the same template but differ in the phone numbers or addresses they contain. For each repetition count, we calculate two loss ranks: one for a canary with a phone number and another for a canary with an address. We only report the lower of these two ranks. The second measure evaluates whether the trained model leaks the secret verbatim during inference. We run inference both with and without prompts. For running inference without prompt, we generate 100,000 synthetic instructions using each trained model and check for the presence of secrets in the generations.

[2] https://pypi.org/project/random-address/

Table 3. Evaluating the leakage of injected canaries. 'Loss Rank' is the rank of a canary among 10,000 instructions that use the same template but different secrets. The rightmost two columns indicate if the trained model leaks the secrets during inference, conducted in two ways. First, we generate 100,000 synthetic instructions without prompts and check for the presence of secrets. Second, we use text preceding the secret as prompt to see if the model's greedy decoding leaks the secret.

Instance-level ε	Repetition	Loss Rank	Leakage (No Prompt)	Leakage (With Prompt)
Non-private	1	50	0/2	0/2
	10	1	1/2	2/2
	100	1	2/2	2/2
$\varepsilon = 2.86$	1	4142	0/2	0/2
	10	1232		
	100	806		
$\varepsilon = 5.94$	1	3566	0/2	0/2
	10	1254		
	100	698		

The hyperparameters used for sampling are the same as those used in Section 4. For running inference with prompts, we utilize the text preceding the secret as the prompt and examine if the model's completion, through greedy decoding, contains the secret. Table 3 presents the leakage of injected secrets. For models trained without DP, the loss ranks of all inserted canaries are alarmingly low, indicating that these models memorize the injected secrets. Furthermore, non-private models leak the secrets when canaries are repeated for 10 or 100 times, whereas private models demonstrate no such leakage.

D. Ablation Studies

D.1. The Impact of Gradient Clipping Thresholds on MAUVE Scores

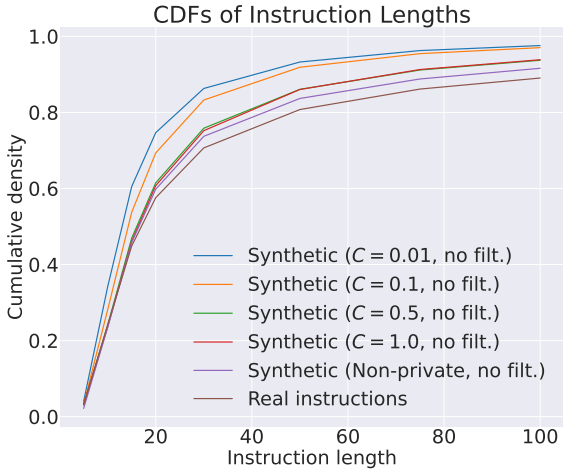


Figure 14. CDFs of instruction length distributions. The generators are trained with different gradient clipping thresholds C. The synthetic instructions are sampled from 7B generative models and are not filtered with Algorithm 2. The privacy budget is $(5.94, 5 \times 10^{-7})$-DP. Small values of C bias the trained models to generate short instructions.

We find that the gradient clipping threshold in DP fine-tuning, denoted as C, influences the quality of the initial synthetic instructions. A smaller value of C makes the fine-tuned generative models bias toward producing shorter instructions. This phenomenon likely occurs because the loss for a sequence is the sum of all token losses in that sequence[3]. As a result,

[3]We normalize the sequence loss by the maximum sequence length to prevent loss explosion. Another common practice is to take the average of token losses as the sequence loss. However, our preliminary experiments suggest that taking the average loss biases short sequences regardless the choice of clipping threshold.

Table 4. The influence of gradient clipping thresholds on MAUVE scores. Models are fine-tuned from LLaMA-7B with $\varepsilon = 5.94$. The instructions are not filtered with Algorithm 2.

	Unigram	Sentence-T5
Non-private	0.980	0.987
$C = 0.01$	0.667	0.843
$C = 0.1$	0.704	0.874
$C = 0.5$	**0.839**	**0.890**
$C = 1.0$	0.819	0.868

longer sequences tend to have higher losses and larger gradients, causing a smaller C to bias the model towards shorter sequences. Conversely, a larger C introduces higher noise variance, which can also degrade the quality of the fine-tuned models. Figure 14 displays the cumulative distribution functions of instruction lengths for varying C values. Table 4 presents the MAUVE scores. We set $C = 0.5$ in the rest of this paper as we find it strikes a balance between these two effects.

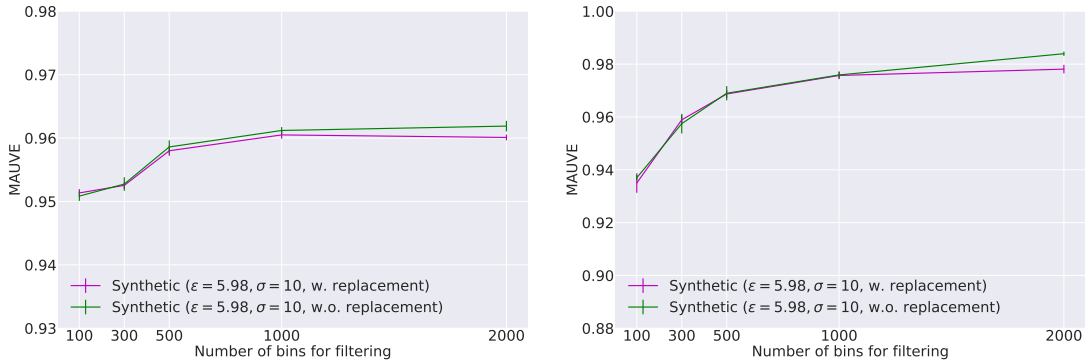
D.2. Larger Pre-trained Models Achieve Better MAUVE Scores

Table 5 presents a comparison between using LLaMA 7B and LLaMA 13B as the pre-trained models. We find that using LLaMA 13B as the pre-train model leads to better MAUVE scores. This aligns with the notion that larger pre-trained models improves DP fine-tuning, as evidenced by recent studies (Ganesh et al., 2023; He et al., 2023; Berrada et al., 2023).

Table 5. MAUVE scores of synthetic datasets generated by 7B and 13B models. The synthetic data is not filtered with Algorithm 2. Fine-tuning 13B models gives better synthetic data.

	Uni-gram	Sentence-T5
Non-private (7B)	0.980	0.987
Non-private (13B)	0.983	0.991
$\varepsilon = 5.94$ (7B)	0.839	0.890
$\varepsilon = 2.86$ (13B)	0.932	0.893
$\varepsilon = 5.94$ (13B)	**0.942**	**0.912**

D.3. Run Algorithm 2 with Fewer Initial Synthetic Instructions



(a) MAUVE-Unigram (b) MAUVE-Sentence-T5

Figure 15. Re-sampling from 500 thousand or one million initial synthetic instructions. For filtering 500 thousand instructions, sampling with replacement is used for $K \geq 1000$ otherwise there are not enough initial samples to select a subset of 180 thousand.

In our experiments in Section 4, we run Algorithm 2 to filter one million synthetic instructions. To investigate the effect of using fewer initial synthetic instructions, we present the MAUVE scores of filtering 500 thousand initial synthetic

instructions in Figure 15. It is noteworthy that for $K = 1000$ and $K = 2000$, we use sampling with replacement during the re-sampling process in Algorithm 2 because otherwise there are not enough initial samples to select from. The results in Figure 15 indicate that for large values of K, using fewer initial synthetic instructions results in worse MAUVE scores.

D.4. Run Algorithm 2 on More Datasets

We use two more datasets to show the wide applicability of Algorithm 2. The datasets are 75,000 IMDB movie reviews[4] and 75,316 PubMed paper abstracts[5] (from August 01 to August 05, 2023). To reduce computational cost, the generators are fine-tuned from GPT2-Large. The hyperparameters for DP fine-tuning follow the best ones in Yue et al. (2022). When Algorithm 2 is applied, we increase the noise multiplier for DP fine-tuning to compensate for the privacy cost of the re-sampling process. We generate 750,000 initial synthetic samples and select a subset that is of the same size as real data. Algorithm 2 consistently improves the quality of initial synthetic data, as indicated by the MAUVE scores in Table 6.

Table 6. MAUVE scores (Sentence-T5) of synthetic IMDB reviews and PubMed abstracts. Generators are fine-tuned from GPT2-Large.

	IMDB	PubMed
Synthetic ($\varepsilon = 3.96$, no filitering)	0.647	0.594
Synthetic ($\varepsilon = 3.93$)	0.825	0.811

D.5. Evaluate SFT Models Using the Non-private Baseline

We evaluate the win rates using a baseline model trained on non-private Chatbot Arena data. The results are presented in Table 7. These findings align with those presented in Table 2 of Section 4.3, where the baseline model is trained on FLAN data. The results show that: 1) the proposed filtering algorithm improves the win-rates; 2) the best-performing DP model are comparable to the non-private baseline.

Table 7. Win-rates with the baseline model trained on non-private Chatbot Arena data.

7B Models	
Chatbot Arena (non-private)	50%
Vicuna-v1.3	46.0% ($\pm$0.20)
Synthetic ($\varepsilon = 5.94$, no filt.)	44.2% ($\pm$0.27)
Synthetic ($\varepsilon = 5.98$)	48.0% ($\pm$0.62)
Synthetic (300K, $\varepsilon = 5.98$)	49.1% ($\pm$0.51)
13B Models	
Vicuna-v1.3	57.7% ($\pm$0.66)
Synthetic (300K, $\varepsilon = 5.98$)	59.2% ($\pm$0.41)

D.6. Evaluate SFT Models on AlpacaEval

In addition to the test set of Chatbot Arena instructions, we also evaluate the models in Section 4.3 on the AlpacaEval dataset (Dubois et al., 2023). AlpacaEval comprises 805 instructions collected from public evaluation sets, including MT-Bench (Zheng et al., 2023c), the self-instruct evaluation set (Wang et al., 2022), the Koala evaluation set (Geng et al., 2023), etc. To prevent data contamination, we again apply an 8-gram deduplication between AlpacaEval instructions and Chatbot Arena instructions. We note that AlpacaEval instructions do not accurately represent real-world user queries to LLMs. For example, the self-instruct evaluation set is generated via prompting GPT-3. Nonetheless, we include the results for a more comprehensive evaluation. We use the same evaluation setup as that used by the AlpacaEval leaderboard[6], which allows us to make direct comparisons with state-of-the-art models listed on the leaderboard. The results are presented in Table 8. When evaluated on AlpacaEval instructions, the best performing DP models slightly underperform Vicuna-v1.3 but are comparable with LLaMA2-Chat.

[4] https://huggingface.co/datasets/stanfordnlp/imdb
[5] https://huggingface.co/datasets/pubmed
[6] https://tatsu-lab.github.io/alpaca_eval/

Table 8. Win-rates evaluated on AlpacaEval. The best performance of DP models are marked in bold. The best DP models are comparable with LLaMA2-Chat but slightly underperform Vicuna-v1.3.

7B Models	
Text-davinci-003	50%
FLAN (non-private)	63.9%
LLaMA2-Chat	71.4%
Vicuna-v1.3	76.8%
Chatbot Arena (non-private)	74.2%
Chatbot Arena ($\varepsilon = 5.94$)	64.9%
Synthetic ($\varepsilon = 5.94$, no filt.)	70.2%
Synthetic ($\varepsilon = 5.98$)	**71.6%**
Synthetic (300K, $\varepsilon = 5.98$)	71.2%
13B Models	
LLaMA2-Chat	81.1%
Vicuna-v1.3	82.1%
Synthetic (300K, $\varepsilon = 5.98$)	**80.6%**

E. Data Cleansing for the LMSYS-Chat-1M Dataset

This section details the data pre-processing procedures applied to the LMSYS-Chat-1M dataset (Zheng et al., 2023a). For the analysis in Appendix B, we utilize a separate dataset: the Chatbot Arena Conversations dataset[7] (Zheng et al., 2023c). The pre-processing steps outlined below are not applied to it.

Specifically, for the LMSYS-Chat-1M dataset, we implement the following sequential pre-processing steps:

Take single-round instructions from English conversations. We first extract user instructions from the dataset, while discarding the responses from LLMs. For the purpose of simplifying our experimental setup, we only consider the instruction from the first round in case of multi-round conversations. Zheng et al. (2023a) provide a language tag for each conversation. We only take instructions from English conversations as English accounts for around 80% of the instructions among over more than twenty languages. Approximately 780,000 instructions reamain after this step.

Remove inappropriate instructions. For each conversation in LMSYS-Chat-1M, Zheng et al. (2023a) utilize OpenAI's moderation API to provide a tag that indicates if a conversation contains inappropriate content. Additionally, they redact real person names in the conversations. We exclude conversations that are either flagged by the moderation API or contain redacted names. Approximately 500,000 instructions remain after this step.

Sequence de-duplication. Users might repeatedly ask the same question to elicit responses from different LLMs. If an instructions appears multiple times, we only keep it once. There are around 310,000 instructions left after this step.

N-gram de-duplication. We divide the instructions into training, validation, and test sets using random splits. To prevent data contamination, we implement an 10-gram de-duplication process. During this process, if an 10-gram is present in multiple instructions, then only the first instruction that has it will be kept. As a result, the dataset is reduced to approximately 232 thousand instructions. Additionally, since we use the Alpaca Eval instructions (Dubois et al., 2023) as one of our evaluation sets, we further exclude any instructions containing 10-grams that appear in the Alpaca Eval instructions. This leads to the removal of fewer than twenty instructions. The dataset is reduced to approximately 220,000 instructions after this step.

Remove unusual repetitions. Upon manually inspecting the remaining instructions, we find unusual repetitions of certain specific patterns. For instance, instructions following these two patterns occur over 16 thousand times.

'Write an instruction of [...] with [...] words in chemical industry'

'Write an article about the [...] [...] words in chemical industry'

After removing those unusual repetitions, there are around 200,000 instructions left. We take 180,000 as the training set,

[7] https://huggingface.co/datasets/lmsys/chatbot_arena_conversations

5,000 as the validation set, and the rest as the test set.

F. Implementation Details

This section documents the implementation details of our experiments. The code to reproduce the main findings in our paper is available at: https://github.com/google-research/google-research/tree/master/dp_instructions.

F.1. Setup for Computing MAUVE

We use the official implementation[8] to compute MAUVE scores. For computing MAUVE with Sentence-T5 embeddings, we follow the default setup to quantize the embeddings into 500 bins. The absolute MAUVE scores are affected by a scaling constant c. Using a larger c would result in a lower score. For computing MAUVE with unigram distributions, we use the default setup $c = 5$. For computing MAUVE with Sentence-T5 embeddings, we set $c = 10$ to better distinguish the scores of different datasets. We note that values of c do not affect the ranking of different datasets.

F.2. Hyperparameters for (DP) Fine-tuning

We employ parameter-efficient fine-tuning as it has been shown to reduce computational cost (Houlsby et al., 2019) and improve the performance in private learning (Yu et al., 2022; Bu et al., 2022; Kurakin et al., 2023). Specifically, we use LoRA fine-tuning (Hu et al., 2022). Following the setup in previous studies (Hu et al., 2022; Li et al., 2022b), we use a reparametrization rank of 8 and only apply LoRA to attention weights. For 7B models, this results in 8.4M trainable parameters. For 13B models, this results in 13.1M trainable parameters. We use the Adam optimizer (Kingma & Ba, 2015) with (β_1=0.9, β_2=0.999). Our experiments use 32 Nvidia A100 40G GPUs. We will open source our code.

Table 9. Hyperparameters for DP fine-tuning. The text in bold indicates the hyperparameters that we eventually use.

LR Schedule	Constant LR
Weight decay	[**0**, 1e-4]
Batch Size	[1024, 2048, **4096**]
Epochs	[3, **10**]
Learning Rate	$[0.5, \mathbf{1}, 3] \times 10^{-3}$
Clipping Threshold	[0.01, 0.1, **0.5**, 1.0]

Table 10. Hyperparameters for non-private fine-tuning. The text in bold indicates the hyperparameters that we eventually use.

LR schedule	Constant LR
Batch size	[32, **64**, 128]
Epochs	[**3**, 10]
Learning rate	$[1, \mathbf{3}, 5, 10] \times 10^{-4}$

Our hyperparameter sweeps for DP fine-tuning and non-private fine-tuning are outlined in Table 9 and Table 10, respectively. The hyperparameter tuning is done with LLaMA 7B. The criteria for the final hyperparameters is achieving the best unigram MAUVE score. Due to computational constraints, we do not perform a full hyperparameter sweep. It is important to note that hyperparameter tuning in DP training does incur an additional privacy cost, albeit modest (Liu & Talwar, 2019; Papernot & Steinke, 2022; Mohapatra et al., 2022; Ding & Wu, 2022; Redberg et al., 2023). To potentially mitigate this overhead, we replicate our hyperparameter tuning process using FLAN instructions as the private dataset. We find that the optimal hyperparameters mirror those of using Chatbot Arena instructions, with the exception of the clipping threshold (1.0 is the optimal value for FLAN). This suggests that tuning hyperparameters on public instructions and transfering the results to private instructions is one viable option for reducing the privacy cost associated with hyperparameter tuning.

F.3. Hyperparameters for Sampling

We employ unconditional sampling to generate synthetic instructions from the fine-tuned models. Specifcially, we use Nucleus Sampling (Holtzman et al., 2019), adhering to its default configuration where the probability cutoff parameter,

[8] https://github.com/krishnap25/mauve

```python
def get_privacy_spent(sampling_prob_dpsgd, running_steps_dpsgd, noise_multiplier_dpsgd, noise_multiplier_histogram):
    from prv_accountant import PRVAccountant
    from prv_accountant.privacy_random_variables import PoissonSubsampledGaussianMechanism, GaussianMechanism

    prv_dpsgd = PoissonSubsampledGaussianMechanism(noise_multiplier=noise_multiplier_dpsgd,
          sampling_probability=sampling_prob_dpsgd)
    prv_histogram = GaussianMechanism(noise_multiplier=noise_multiplier_histogram)

    accountant = PRVAccountant(
          prvs=[prv_dpsgd, prv_histogram],
          max_self_compositions=[running_steps_dpsgd, 1],
          eps_error=0.01,
          delta_error=1e-8,
    )

    eps_lower_bound, eps_estimated, eps_upper_bound = accountant.compute_epsilon(delta=5e-7,
          num_self_compositions=[running_steps_dpsgd, 1])

    return eps_upper_bound
```

Figure 16. The implementation of our privacy accounting function. We use the PRV accounting package provided by Gopi et al. (2021).

top_p is set as 0.95. Following previous work Yue et al. (2022); Kurakin et al. (2023), the sampling temperature T is set as 1.0. In our preliminary experiments, we set T=0.7 and observe a significant deterioration in the MAUVE scores. The maximum sequence length for the generated instructions is 1024 tokens.

For supervised fine-tuning, we follow the prompt for Vicuna. The prompt template is *'A chat between a curious user and an artificial intelligence assistant. The assistant gives helpful, detailed, and polite answers to the user's questions. USER: [Instruction] ASSISTANT:'*. The maximum sequence length for the generated responses is set as 2048 tokens. During inference, we follow the default setup of Vicuna[9]. Specifically, sampling temperature T and probability cutoff parameter top_p are set as 0.7 and 1.0, respectively.

F.4. Code Snippet for Privacy Accounting

The Python code of our privacy accounting function is in Figure 16. We use the Privacy Random Variable (PRV) accountant (Gopi et al., 2021). For running DP-Adam for 10 epochs with a batchsize of 4096 and $(5.94, 5 \times 10^{-7})$-DP, we set the noise multiplier for DP-Adam as 0.81. For $(2.86, 5 \times 10^{-7})$-DP, we set the noise multiplier as 1.11. The default noise multiplier for releasing the histogram in Algorithm 2 is 10.0.

F.5. Details for Annotating with GPT-3.5-Turbo

We use gpt-3.5-turbo-0301[10] to create responses for all instructions. We use 20 parallel threads to query the OpenAI API. Annotating 100 thousand instructions takes approximately 12 hours and $80. The maximum response length is 2048 tokens. The sampling temperature T is set as 0.7. The probability cutoff parameter top_p is set to 0.95.

F.6. Hyperparameters for Proximal Policy Optimization

The baseline model[11] is a 1.3B model fine-tuned on the OpenOrca dataset. The reward model[12] comprises 304M parameters and is trained on several public human preference datasets, such as the annotated version of the TL;DR dataset featuring Reddit posts (Völske et al., 2017; Stiennon et al., 2020). The maximum sequence length for generated responses is set as 256 tokens. The sampling temperature and the probability cutoff parameter top_p are both set to 1.0 during training. Our PPO implementation is based on the Transformer Reinforcement Learning (TRL) library. We follow the default setup for PPO training[13] for most of the hyperparameters. The KL penalty coefficient is set as 0.2 and the number of optimisation steps per batch is 4. We use a batchsize of 128 and a learning rate of 1.41×10^{-5}. All models are trained for 2500 batches of samples.

[9] https://github.com/lm-sys/FastChat/tree/main
[10] https://platform.openai.com/docs/models/gpt-3-5
[11] https://huggingface.co/Open-Orca/oo-phi-1_5
[12] https://huggingface.co/OpenAssistant/reward-model-deberta-v3-large-v2
[13] https://github.com/huggingface/trl/blob/main/trl/trainer/ppo_config.py

G. Additional Figures

We put some figures here because of the space constraints. In Figure 17, we present random samples from two particular clusters of initial synthetic instructions: one cluster receiving the highest number of votes from real data and the other receiving the lowest. In Figure 18, we compare the sequence length distribution of Chatbot Arena instructions with the length distribution of responses from GPT-3.5-Turbo.

The cluster with the highest vote count.

1. How to find slope and y-intercept from equation of a line using python?
2. Code in python for calculating the intial investment value for a minimum annual 9\% returns given the investment value and the end year return value.
3. please provide python code to plot bar graph

The cluster with the least vote count.

1. ข่าวล่ารหุา มาวี่โกแลกกิจกลาง รีบี้ร อิสร์วี่ ฉกรรจ์ วิสชุน และอันตรานะ ไม่ยงวนยบงกรวดรอยยางจบ กว่าดูการจัดลงเสียง กล้าขาค่าร้ายวิวเวียน
2. ##\
3. 欢迎 文库接入员!

Figure 17. Random synthetic instructions from clusters that have the highest and lowest vote counts.

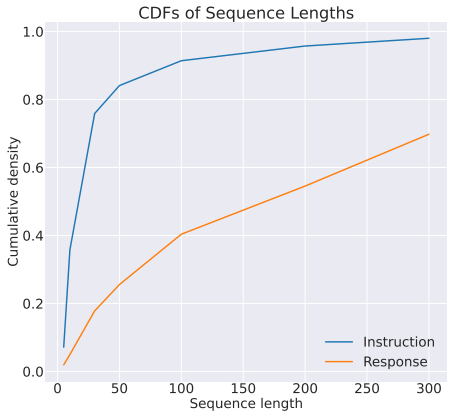


Figure 18. Cumulative distribution functions of sequence length distributions. The responses are generated by GPT-3.5-Turbo. Responses are typically much longer than instructions.