PrE-Text: Training Language Models on Private Federated Data in the Age of LLMs

Charlie Hou [1] **Akshat Shrivastava** [2] **Hongyuan Zhan** [2] **Rylan Conway** [2] **Trang Le** [2]
Adithya Sagar [2] **Giulia Fanti** [1] **Daniel Lazar** [3]

Abstract

On-device training is currently the most common approach for training machine learning (ML) models on private, distributed user data. Despite this, on-device training has several drawbacks: (1) most user devices are too small to train large models on-device, (2) on-device training is communication- and computation-intensive, and (3) on-device training can be difficult to debug and deploy. To address these problems, we propose Private Evolution-Text (PrE-Text), a method for generating differentially private (DP) synthetic textual data. First, we show that across multiple datasets, training small models (models that fit on user devices) with PrE-Text synthetic data outperforms small models trained on-device under practical privacy regimes ($\epsilon = 1.29$, $\epsilon = 7.58$). We achieve these results while using $9\times$ fewer rounds, $6\times$ less client computation per round, and $100\times$ less communication per round. Second, finetuning large models on PrE-Text's DP synthetic data improves large language model (LLM) performance on private data across the same range of privacy budgets. Altogether, these results suggest that training on DP synthetic data can be a better option than training a model on-device on private distributed data. Code is available at https://github.com/houcharlie/PrE-Text.

1. Introduction

In many language applications, such as mobile keyboard autocompletion (McMahan et al., 2017b), or instruction-following large language models (Yu et al., 2024), training a model on private user data can significantly improve model performance. However, user data is often sensitive, necessitating the use of algorithmic techniques for protecting privacy. Federated Learning (FL) (McMahan et al., 2017a) is a prominent technique that trains models on user devices (we call this on-device training) and aggregates the resulting model updates at a central server. Recent works have shown that FL combined with differential privacy (DP) (Dwork, 2006)—a combination we refer to as *DP-FL*—can protect privacy while also improving model performance in user applications (McMahan et al., 2017b; Kairouz et al., 2021b;a; Nguyen et al., 2022; Xu et al., 2023b).

On-device training or federated learning has several drawbacks. (1) Due to limited on-device storage and computation, client devices cannot be used to train large language models (LLMs) (Radford et al., 2019; Touvron et al., 2023). As LLMs become more critical in many use-cases, this becomes more limiting (Charles et al., 2023). (2) On-device training can have high communication and computation costs for clients (Cai et al., 2022). Indeed, there is a large body of literature studying how to improve the efficiency of on-device training (Wang et al., 2020; Li et al., 2020; Karimireddy et al., 2020; Hou et al., 2021a; Wang et al., 2021; Mishchenko et al., 2022; Sadiev et al., 2022; Grudzień et al., 2023). (3) It is difficult to deploy and debug (Augenstein et al., 2019), requiring extensive infrastructure investment (Tensorflow, 2018; Zhao et al., 2023; PyTorch, 2024).

An alternative paradigm: Train or finetune on differentially private (DP) synthetic data. We propose to have the central server first generate DP synthetic data from private client data, then centrally finetune a pretrained language model on that private synthetic data. As in on-device training, clients send DP information to the server; this is used by the server to generate high-quality synthetic data. Unlike on-device training, clients do not need to run training steps for the downstream model. Finetuning on DP synthetic data located on-server (1) eliminates the model size constraints of on-device training, (2) is easier to debug as we can observe the training process without compromising DP, and (3) does not require new training infrastructure, unlike on-device training. Furthermore, DP synthetic data located on

[1]Department of Electrical and Computer Engineering, Carnegie Mellon University [2]Meta [3]Coldrays. Correspondence to: Charlie Hou <hou.charlie2@gmail.com>. Charlie Hou and Daniel Lazar partially worked on this paper while at Meta.

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

Table 1. PrE-Text provides the privacy guarantees of on-device training while (1) being much cheaper in communication and computation, and (2) being much easier to practitioners to deploy in real systems.

Method	Train LLMs? [(a)]	Is communication cheap? [(b)]	Is computation cheap? [(c)]	Easy to deploy? [(d)]	Privacy-preserving?
Private on-device training DP-FedAvg (McMahan et al., 2017b) DP-FTRL (Kairouz et al., 2021b)	✗	✗	✗	✗	✓
Centralized non-private training	✓	✓	✓	✓	✗
PrE-Text (proposed)	✓	✓	✓	✓	✓

[(a)] **Training LLMs.** On-device finetuning of large models (like LLMs) is not feasible because LLMs are too big. PrE-Text allows us to finetune LLMs because the resulting synthetic data is located on-server.
[(b)] **Communication cost.** PrE-Text required 100x less communication per round (and 9x fewer rounds) in our experiments.
[(c)] **Client computation cost.** PrE-Text required 6x less client computation per round (and 9x fewer rounds) than on-device training in our experiments. This comes at the cost of more server-side computation resources, which is often much less constrained.
[(d)] **Practicality.** PrE-Text produces synthetic data on-server, which allows practitioners to see the training process end-to-end; this improves debuggability (Augenstein et al., 2019). Furthermore, the synthetic data can be reused an unlimited number of times without incurring additional privacy cost.

server can be reused an unlimited number of times to train many models without incurring additional privacy cost, due to the post-processing property of DP (Dwork, 2006).

Unfortunately, existing techniques for generating DP synthetic language data from federated clients are too low-quality to train or fine-tune a language model (Augenstein et al., 2019). We fill this gap by leveraging Private Evolution (PE) (Lin et al., 2023), a recent algorithmic breakthrough in DP synthetic data. PE is a framework for generating realistic DP synthetic image data (Lin et al., 2023), which achieves high scores in realism metrics like FID (Heusel et al., 2017). However, Lin et al. (2023) do not apply PE to text, nor do they show that training on DP synthetic data is a competitive alternative to direct DP training (DP-FL or DP-SGD (Abadi et al., 2016)) on private data. Our work utilizes PE in the natural language setting (which is a nontrivial adaptation) as part of our overall algorithm, then demonstrates through extensive experiments that the resulting synthetic data can produce better models than DP-FL at a fraction of the cost.

We list our contributions below:

(1) PrE-Text (Private Evolution-Text) algorithm. We propose PrE-Text, a new algorithm for DP synthetic text generation. We build on the following insights: (1) PE must generate variations of samples (e.g., similar images). We adapt this requirement to the language domain by carefully utilizing mask-filling models (Devlin et al., 2018; Lewis et al., 2019) instead of the diffusion models they used for images. (2) PE alone does not generate enough high-quality synthetic data to effectively finetune an LLM. Hence, we add a post-processing phase, in which we use the outputs of PE to seed high-quality LLMs trained on public data to generate more similar text.[1] Done carefully, this can generate orders of magnitude more useful training data, greatly aiding generalization (all without incurring more privacy cost, due to the post-processing property of DP (Dwork, 2006)).

(2) Experimental results. We produce high-quality DP synthetic language data using PrE-Text, and demonstrate its superiority over other methods in two major settings:

a) Models served on-device. These models are small enough to fit on user devices. We show that in this setting, models trained on synthetic data produced by PrE-Text achieve similar or better performance to models trained on-device (at $\epsilon = 1.29$ and $\epsilon = 7.58$, these privacy levels are standard (McMahan et al., 2017b)), with $\sim$100$\times$ less communication per round, $\sim$6$\times$ lower client computation per round, and 9$\times$ fewer rounds. We show that in these important privacy regimes, PrE-Text outperforms on-device training for training next-token-prediction models.

b) Models served on-server. These are the models that are too large (in our setting, LLMs) to be served on user devices.

[1] In concurrent work, Xie et al. (2024) extend PE to the text domain using similar techniques to ours, developing a new algorithm called Aug-PE. Some differences: (1) Their focus is purely on generating better DP synthetic text data, whereas ours is on understanding whether DP synthetic data can take the place of on-device learning. To this end, they study the centralized setting, while we study the federated setting. (2) PrE-Text uses the original PE as a subroutine in a larger algorithm, whereas Aug-PE is Private Evolution with core components rewritten. (3) Their approach relies on a closed-source model (ChatGPT) for consistent improvements over baselines, whereas ours uses open-source models (LLaMA). This matters because prompts for synthetic data generation that work on ChatGPT did not work well on LLaMA, and the OpenAI API policy might not allow training on outputs (OpenAI, 2024).

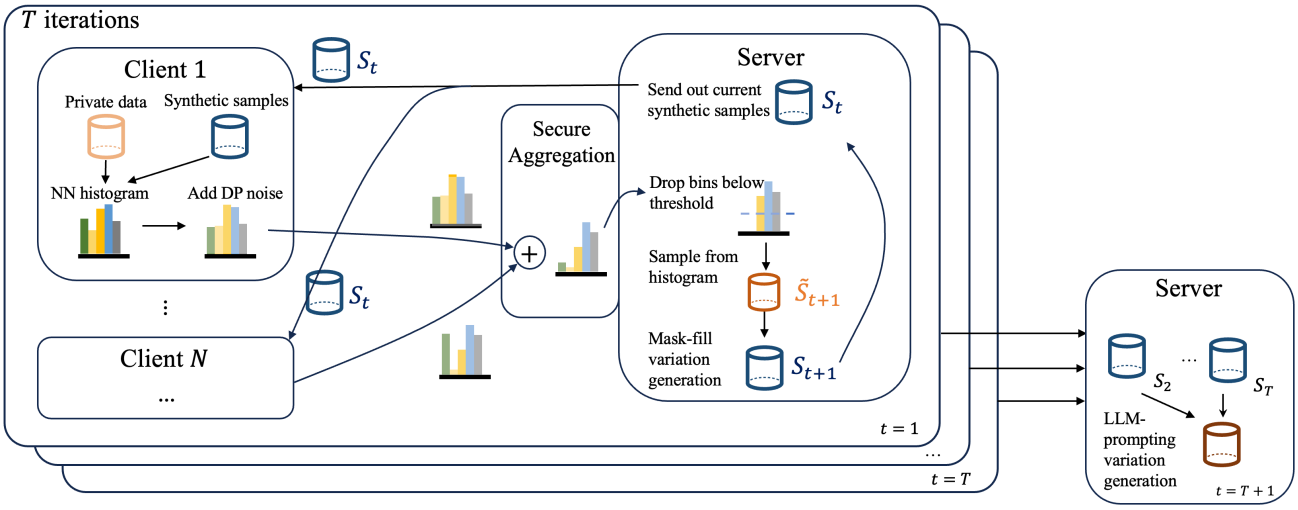

Figure 1. A high-level description of PrE-Text. PrE-Text consists of two main phases: (1) (iterative) DP synthetic seed collection, (2) (single-shot) synthetic seed expansion. A detailed description of steps in the diagram is given in Section 3.

We demonstrate that in this setting, large models finetuned on synthetic data produced by PrE-Text perform better in next-token-prediction than non-finetuned pretrained LLMs. To the best of our knowledge, we are the first to privately finetune an LLM in the federated setting without requiring the model to be held on user devices. With major LLM providers running out of useful *public* data to train on (Xu, 2022; Seetharaman, 2024; Needleman, 2024; Tong et al., 2024), PrE-Text offers a promising path forward. PrE-Text can use *private* client data in a way that is both mindful of user hardware constraints and is also privacy-compliant.

2. Preliminaries

In this section, we provide formal definitions for our setting.

Definition 2.1 (Neighboring datasets)**.** Two federated datasets X, X' are said to be neighboring (denoted $X \sim X'$) if they differ at most with respect to only one user's data (i.e. X has an extra user compared to X' or vice versa). Note that we consider a *user-level* notion of neighboring datasets (McMahan et al., 2017b).

Definition 2.2 (Differential Privacy)**.** A randomized algorithm $\mathcal{A}$ is (ϵ, δ)-differentially private (DP) if for any pair of neighboring datasets X, X' and for all subsets E of outputs,

$$\Pr[\mathcal{A}(X) \in E] \leq e^{\epsilon} \Pr[\mathcal{A}(X') \in E] + \delta. \quad (1)$$

In this work, we use the Gaussian Mechanism (Dwork, 2006) for DP, which adds Gaussian noise of a specific scale to released statistics. The required scale of noise depends on the *sensitivity* of the statistical query we wish to release. When defining the Gaussian noise, we use I_n to represent the identity matrix of size $n \times n$.

Definition 2.3 (ℓ_2 sensitivity (Dwork et al., 2014))**.** Let $g : X \to \mathbb{R}^p$ be a vector-valued function operating on datasets. Let X, X' be neighboring datasets. The ℓ_2-sensitivity of g is defined as $\Delta g := \max_{X \sim X'} \|g(X) - g(X')\|_2$.

2.1. Problem formulation

Private clients setup. We consider a setting where a central server wishes to learn a model $\mathcal{M}$ from N user devices (or, "clients"). Client i has the language dataset $\mathcal{D}_i$ which consists of $|S_i|$ language samples. M of these clients, $\mathcal{C}_{\text{test}} \subset [N]$ are considered test clients, and we cannot access their data during synthetic data generation and the training process. The remaining $N - M$ clients, $\mathcal{C}_{\text{train}} \subset [N]$, are considered training clients, and their data can be accessed during model training and/or synthetic data generation. We assume that the client language datasets are drawn from a distribution of possible client datasets $\hat{\mathcal{D}}$, so each $\mathcal{D}_i$ is drawn independently from $\hat{\mathcal{D}}$, $\mathcal{D}_i \sim \hat{\mathcal{D}}$.

We divide the space of models into two: on-device models, which can fit on a client device, and on-server models, which cannot. We assume that large language models (LLMs) and other large foundation models are on-server models.

Server setup. The server has access to pretrained LLMs (for example, the LLaMA models in our setting (Touvron et al., 2023)) which were trained only on public data.

Task. We focus on the language modeling task, where a language model predicts token s_k from the previous tokens $s_0, \cdots, s_{k-1}$ for each text sample. The server's final goal is to learn a language model that performs well on next-token-prediction on the private test dataset $\mathcal{C}_{\text{test}}$.

Algorithm 1 `PrE-Text`

Input: Number of iterations: T
1: Number of generated samples: N_{syn}
2: Initial population: S_1
3: **Output:** Synthetic data S_{syn}
4: **for** $t \leftarrow 1 \dots T$ **do**
5: // SEE ALGORITHM 2
6: $\text{hist}_t \leftarrow \texttt{Histogram}(S_t)$
7: $P_t \leftarrow \text{hist}_t / \text{sum}(\text{hist}_t)$
8: $S'_t \leftarrow$ draw N_{syn} samples from P_t
9: $S_t \leftarrow \texttt{Variation}(S'_t)$
10: **end for**
11: $S_{\text{syn}} \leftarrow \texttt{Expand}(\cup_{t=1}^{T} \texttt{Set}(S'_t))$
12: **return** S_{syn}

Algorithm 2 `Histogram`

1: **Settings:** Private samples: Clients $\{C_i\}_{i \in \mathcal{C}_{\text{train}}}$
2: Noise multiplier for histogram: σ
3: Number of generated samples: N_{syn}
4: Threshold for histogram: H
5: Distance function: $d(\cdot, \cdot)$
6: **Input:** Generated samples $S = \{z_i\}_{i=1}^{N_{\text{syn}}}$
7: **Output:** Nearest neighbors histogram on S
8: **for** $i \in \mathcal{C}_{\text{train}}$ **do**
9: $\text{hist}_i \leftarrow [0, \dots, 0]$
10: **for** $x_{\text{priv}} \in \mathcal{D}_i$ **do**
11: $l \leftarrow \text{argmin}_{j \in [N_{\text{syn}}]} d(x_{\text{priv}}, z_j)$
12: $\text{hist}_i[l] \leftarrow \text{hist}_i[l] + 1$
13: **end for**
14: $\text{hist}_i \leftarrow \text{hist}_i + \mathcal{N}(0, (\sigma^2 / |\mathcal{C}_{\text{train}}|) I_{N_{\text{syn}}})$
15: **end for**
16: $\text{hist} \leftarrow \sum_{i \in \mathcal{C}_{\text{train}}} \text{hist}_i$
17: $\text{hist} \leftarrow \max(\text{hist} - H, 0)$ (elementwise subtraction)
18: **return** hist

Privacy and threat model. We consider an honest-but-curious threat model (Nguyen et al., 2022). Using secure aggregation (Bonawitz et al., 2016), the server does not see individual client uploads, but rather the aggregated upload across clients. By adding DP noise, clients prevent the server from inferring any single client's data from the aggregated upload (which contains an aggregated amount of noise). The server then aims to learn an (ϵ, δ)-DP language model (where the notion of DP is user-level distributed DP[2]).

[2]Note that technically, using the Gaussian Mechanism together with secure aggregation requires discretization as secure aggregation uses modular arithmetic (Kairouz et al., 2021a; Bagdasaryan et al., 2021; Agarwal et al., 2021). This is an important consideration when practically deploying end-to-end DP applications relying on secure aggregation and the Gaussian mechanism.

3. PrE-Text

The main intuition of PrE-Text (and PE) is that public foundation models should be capable of producing samples that are similar to the private data with some non-negligible probability. Therefore, to generate DP synthetic data similar to private data, we steer the foundation model (privately) towards the user data in a multi-round process. Briefly, we make several important changes to the PE algorithm: (1) we adapt it from the image setting to the text setting; (2) we exploit synthetic data from the *entire* PE process, rather than only the last round; and (3) we add a post-processing phase that utilizes the output of PE as seeds for another LLM. We provide pseudocode in Algorithm 1, and highlight the new contributed steps in color. We now explain Algorithm 1.

(1) Population of samples. We start with an initial population of samples S_1 (Line 2). These samples can come from many different sources as long as they do not contain private information: for example public samples collected from the internet or samples randomly generated by a public generative foundation model.

(2) Clients vote for the best synthetic samples. For round $t \geq 1$, we determine which of the generated samples S_t represent the private samples the best. We send all the generated samples S_t to each client. Each client counts for each generated sample $s \in S_t$ how many private samples had s as their nearest neighbor in S_t. The higher this count is, the "better" a generated sample is. Thus, each client produces a nearest neighbors histogram with $|S_t|$ entries (Line 9). We determine nearest neighbor according to a distance function $d(y, z) = \|\Phi(y) - \Phi(z)\|_2$, where Φ is an embedding model.

Lookahead. To more accurately assess the closeness of a synthetic sample to a private sample, we amend the distance function from $d(y, z) = \|\Phi(y) - \Phi(z)\|_2$ to $d(y, z) = \|\Phi(y) - \frac{1}{K}\sum_{i=1}^{K} \Phi(z^i)\|_2$, where $z^0, \dots, z^K$ are K variations of z produced by using `Variation`. Lin et al. (2023) also uses this modification. Instead of sending the actual generated samples directly to all the clients, we send $\frac{1}{K}\sum_{i=1}^{K} \Phi(z^i)$ to all the clients for every i for their nearest neighbors calculation (Line 11).

(a) DP Noise. In Line 14 each client adds noise to their nearest neighbors histogram to ensure DP. We compute client-level sensitivity assuming a known upper bound on the number of samples per client (e.g., via thresholding).

(b) Federated Secure Aggregation. In Line 16 we securely aggregate the histograms across the users. Because the generated samples given to all the clients are the same, we sum the histograms using secure aggregation (Bonawitz et al., 2016) to get an aggregate histogram.

(c) Thresholding. When we generate many samples, the

majority of the probability mass of the histogram will be noise. We improve the signal-to-noise ratio by thresholding the histogram at H in Line 17 (Lin et al., 2023).

(3) Use votes to choose the surviving samples. We sample from the nearest neighbors histogram to produce surviving generated samples in Line 8, S'_{t+1}. This new list of generated samples (there may be duplicates) tends to give more representation to generated samples that had more private samples close to them.

(4) Produce variations of surviving samples. We use `Variation` to generate a variation of the surviving samples S'_{t+1} as the new population of samples (Line 9). In PE, this was accomplished using diffusion models, which cannot be used for text. We instead implement `Variation` as follows. For each sample in S'_t, we produce a variation of it by masking MASK% of the tokens randomly, then filling in those masked tokens in with a masked language model. The resulting sample is then masked and filled-in again. This mask-fill process happens W_{steps} times before returning the variation. We use RoBERTa-large (Liu et al., 2019) as the masked language model.

(5) Making efficient use of iterates. We make several major modifications to the core Private Evolution algorithm to improve our usage of the iterates. First, Lin et al. (2023) use the final S_T (Line 9) as the synthetic dataset. However, S'_T contains more information about the private dataset than S_T because `Variation` destroys information. Therefore, we use S'_T instead of S_T. Second, we find that S'_t for $t = 1, \ldots, T$ all have valuable synthetic samples and show significant diversity between iterations. Therefore, we choose to utilize $\bigcup_{t=1}^{T} \texttt{Set}(S'_t)$ (Line 11) instead of just S'_T. This more effectively utilizes the privacy budget.

(6) Post-processing: Use LLM to expand the DP seed set. PE originally used the final S_T as the target synthetic data. However, we find that these samples are not high enough quality to fine-tune an LLM. We instead use `Expand` to generate more samples similar to the synthetic samples $\bigcup_{t=1}^{T} \texttt{Set}(S'_t)$, which we use as our (DP) seed set to prompt the LLM. `Expand` utilizes the synthetic data generation capabilities of large language models (LLMs) to generate a larger and more useful synthetic dataset. Note that by the post-processing property of differential privacy (Dwork, 2006), `Expand` will not leak any additional privacy. Next we describe how `Expand` works.

Inspired by highly successful synthetic text generation (Taori et al., 2023; Wang et al., 2022; Honovich et al., 2022; Rozière et al., 2023), we generate synthetic text by using large foundational language models. We follow a process similar to Honovich et al. (2022): for each synthetic sample to generate, we randomly choose three text samples to emulate from $\bigcup_{t=1}^{T} \texttt{Set}(S'_t)$ (our DP seed set), and ask the LLM to generate a similar sample. We use open-source LLaMA-2-7B (Touvron et al., 2023) as our large language model. We provide the full prompt (Figure 3).

3.1. Privacy analysis

As noted in Lin et al. (2023), the only function that utilizes private information is Algorithm 2. The DP histogram (Line 16) contains private information. As in Lin et al. (2023), we use the Gaussian mechanism with constant noise multiplier σ each time we receive a histogram. Since this is a Gaussian mechanism, we can use the moments accountant from the Opacus library (Yousefpour et al., 2021). Details can be found in Appendix D.1.

4. Experiments

Models. We use RoBERTa-large (Liu et al., 2019) for mask-filling. We use all-MiniLM-L6-v2 for text embeddings. We use DistilGPT2 (Sanh et al., 2019) to evaluate our methods. Finally, we use LLaMA-2-7B (Touvron et al., 2023) for synthetic seed expansion.

Datasets. We produce three federated private datasets from the c4-English (Raffel et al., 2019) (c4-en): JOBS, FORUMS, and MICROBLOG, which are subsets of c4-en. In these datasets, the federated datasets are uniformly randomly partitioned among clients. We also produce another federated private dataset CODE, a question-and-answer dataset focused on coding and technical topics. For all training datasets, there are 1250 clients. The evaluation sets are created from a held-out portion of the data. For the initial population used in PrE-Text, we use a subset of c4-en that is not part of any of the private datasets. More details on the datasets are provided in Appendix C.

Note that many LLMs do not document what datasets were used in their pretraining, which makes it difficult to prevent contamination. Even text released after the release of the model may be contaminated, as it may have been AI-generated. We used the most recent large-scale dataset we could find (though it was released before the release of LLaMA-2) that is (a) compliant with terms of service (many sources of recent text data have closed their APIs for ML training) and (b) readily accessible. Systematically detecting dataset contamination is an important open problem in LLM research (Gunasekar et al., 2023).

Task. We focus on the language modeling task, and report evaluation loss (cross-entropy) and accuracy. We consider two experimental settings: (small) models stored on-device, and (large) models stored on-server.

Baselines. We compare PrE-Text to several baselines.

(1) $\epsilon = 0$ baselines: Our first two baselines, **c4-only** and **Expand-only**, give lower bounds on performance by not

using the private data at all, and relying only on public data. c4-only is a DistilGPT2 model finetuned on a subset of c4-en that came from website sources not represented in any of the private data. As Xu et al. (2023c) found, finetuning on c4-en improved privacy-utility tradeoff greatly in next-token prediction, so this is an important baseline. Expand-only is a DistilGPT2 model finetuned on the subset of c4-en used in the c4-only baseline, expanded to 2 million samples using `expand`.

(2) $\epsilon = \infty$ baselines: Our next baseline provides an upper bound on model performance, obtained by ignoring privacy constraints. To this end, we evaluate **Expand-private**. This is a DistilGPT2 model finetuned on (1) the subset of c4-en used in c4-only baseline, and (2) the private dataset expanded to 2 million samples using `expand`. We found that this performs better than a model finetuned on only the private dataset itself.

(3) On-device baselines: We next evaluate two representative on-device training baselines that use DP optimization to provide a privacy guarantee: **DP-FedAvg** (McMahan et al., 2017b) and **DP-FTRL** (Kairouz et al., 2021b), which are two of the most widely-used algorithms for private on-device training in practice. Specifically, we first finetune DistilGPT2 on the subset of c4-en used in the c4-only baseline, and then finetune it further using DP-FedAvg or DP-FTRL (which are on-device training methods). We use the DP-FTRL-TreeRestart variant of DP-FTRL, as we consider full participation in each communication round.

(4) Text-to-text privatization baseline: We finally compare PrE-Text against **DP-Prompt**, a different approach for generating DP synthetic data with paraphrasing. In this approach, clients hold an LLM on-device and release privacy-preserving paraphrases of their text directly to the server. The representative method we use here is DP-Prompt (Utpala et al., 2023). We use the same prompt and model (flan-t5-3b) as Utpala et al. (2023). Note that these methods cannot take advantage of secure aggregation (text cannot be summed) which necessitates adding much more noise to the privatized text to guarantee user-level privacy. We first finetune a DistilGPT2 model on a subset of c4-en used in the c4-only baseline, and then finetune it further on the privatized text received by the server.

4.1. Models stored on-device

We consider a setting where users do not send data directly to a server-side LLM and instead keep a small model on-device for inference. We use DistilGPT2 as our representative example of a small model, with 82M parameters.

Experimental setup. For **PrE-Text**, we generate a synthetic dataset of 2 million samples. We first finetune DistilGPT2 on the subset of c4-en used in the c4-only baseline, and then finetune it further on the synthetic dataset generated by PrE-Text. We compare against all the baselines mentioned at the beginning of the section. For more details on how we instantiated each method including hyperparameter grids, see Appendix D.2.

Results. We present our results in Table 2 and Table 3. We find that PrE-Text outperforms all other baselines at $\epsilon = 1.29$ and $\epsilon = 7.58$. As a sanity check, it also performs better than the $\epsilon = 0$ baseline and worse than the $\epsilon = \infty$ baseline. The results show that in practical privacy regimes, PrE-Text outperforms private on-device training. As synthetic data generation methods improve (for example, better prompts and models for `expand`), synthetic data generation may continue to greatly improve as a strategy for learning from private federated datasets.

Efficiency differences. We compare the efficiency of DP-FL vs PrE-Text in our experimental setup, demonstrating that PrE-Text is much more efficient in terms of computation and communication in our setting.

(1) Communication cost. DP-FL requires each client to download and upload the model (DistilGPT2) to/from the server. DistilGPT2 has a size of 82M parameters, which means clients are downloading and uploading 82M floats each round. On the other hand, PrE-Text requires clients to download at most 2048 embedding vectors of size 384 representing the synthetic samples each round. This means each client downloads around 800K floats. On upload, clients upload at most 2048 floats (the size of the histogram). So client download cost is 100× cheaper with PrE-Text per round, and upload cost is 41000× cheaper with PrE-Text per round. In addition, PrE-Text uses 9× fewer rounds than the on-device baselines. Conservatively, this is at least a 100× improvement in communication cost per round when using PrE-Text (possibly more, given that upload speeds can be 15× slower than download (speedtest.net, 2023)). The embedding model we use, miniLM-L6-v2 (Reimers & Gurevych, 2019), is itself only 10M floats, and only needs to be downloaded once per user, and can be done offline.

(2) Client computation cost. PrE-Text is also much more computationally efficient for clients. We assume that user devices only have access to CPU computation, as most smartphones do not have GPUs (or have fairly limited GPUs). When tested on a VM with five Intel(R) Xeon(R) Gold 6248 CPU @ 2.50GHz, training with DistilGPT2 requires 3 seconds per sample to train while the client computation for PrE-Text (nearest neighbors calculation and client embedding generation) requires less than 0.5 seconds per sample. This computation gain comes from the fact that clients perform inference (not training), which requires fewer operations and can be sped up (Kwon et al., 2023; Reimers & Gurevych, 2019). This gives at least a 6× improvement in computation cost per round for clients with PrE-Text. In

Table 2. We compare the next token prediction accuracies (higher is better) achieved by PrE-Text against DP-Prompt/DP-FedAvg/DP-FTRL, under $(1.29, 3 \times 10^{-6})$-DP and $(7.58, 3 \times 10^{-6})$-DP. We also compare these methods against baselines with $\epsilon = 0, \epsilon = \infty$. We see that PrE-Text outperforms the alternatives at $\epsilon = 1.29$ and $\epsilon = 7.58$. The error bars are stderr.

Method	Privacy	JOBS (↑)	FORUMS(↑)	MICROBLOG (↑)	CODE (↑)
c4-only	$\epsilon = 0$	0.695 ± 0.000	0.650 ± 0.000	0.658 ± 0.000	0.622 ± 0.000
Expand-only		0.702 ± 0.000	0.661 ± 0.000	0.669 ± 0.000	0.635 ± 0.001
DP-Prompt	$\epsilon = 1.29$	0.673 ± 0.000	0.636 ± 0.000	0.642 ± 0.000	0.599 ± 0.000
DP-FedAvg		0.701 ± 0.000	0.663 ± 0.000	0.665 ± 0.000	0.636 ± 0.000
DP-FTRL		0.703 ± 0.000	0.665 ± 0.000	0.667 ± 0.000	0.643 ± 0.000
PrE-Text		$\mathbf{0.718 \pm 0.000}$	$\mathbf{0.672 \pm 0.000}$	$\mathbf{0.680 \pm 0.000}$	$\mathbf{0.644 \pm 0.001}$
DP-Prompt	$\epsilon = 7.58$	0.672 ± 0.000	0.637 ± 0.000	0.642 ± 0.000	0.598 ± 0.000
DP-FedAvg		0.703 ± 0.000	0.666 ± 0.000	0.666 ± 0.000	0.643 ± 0.000
DP-FTRL		0.704 ± 0.000	0.667 ± 0.000	0.668 ± 0.000	0.644 ± 0.000
PrE-Text		$\mathbf{0.721 \pm 0.000}$	$\mathbf{0.673 \pm 0.000}$	$\mathbf{0.680 \pm 0.000}$	$\mathbf{0.652 \pm 0.000}$
Expand-private	$\epsilon = \infty$	0.730 ± 0.000	0.684 ± 0.000	0.689 ± 0.000	0.687 ± 0.000

Table 3. We compare the next token prediction (cross-entropy) losses (lower is better) achieved by PrE-Text against DP-Prompt/DP-FedAvg/DP-FTRL, under $(1.29, 3 \times 10^{-6})$-DP and $(7.58, 3 \times 10^{-6})$-DP. We also compare these methods against baselines with $\epsilon = 0, \epsilon = \infty$. We see that PrE-Text outperforms the alternatives at $\epsilon = 1.29$ and $\epsilon = 7.58$. The error bars are stderr.

Method	Privacy	JOBS (↓)	FORUMS(↓)	MICROBLOG (↓)	CODE (↓)
c4-only	$\epsilon = 0$	1.781 ± 0.004	2.154 ± 0.007	2.103 ± 0.004	2.492 ± 0.007
Expand-only		1.611 ± 0.000	1.883 ± 0.000	1.858 ± 0.000	2.158 ± 0.002
DP-Prompt	$\epsilon = 1.29$	1.799 ± 0.003	2.049 ± 0.001	2.055 ± 0.000	2.312 ± 0.003
DP-FedAvg		1.644 ± 0.000	1.888 ± 0.000	1.916 ± 0.000	2.143 ± 0.000
DP-FTRL		1.594 ± 0.000	1.850 ± 0.000	1.874 ± 0.000	2.007 ± 0.000
PrE-Text		$\mathbf{1.482 \pm 0.001}$	$\mathbf{1.779 \pm 0.001}$	$\mathbf{1.770 \pm 0.000}$	$\mathbf{2.004 \pm 0.010}$
DP-Prompt	$\epsilon = 7.58$	1.801 ± 0.002	2.047 ± 0.000	2.055 ± 0.001	2.317 ± 0.001
DP-FedAvg		1.598 ± 0.000	1.854 ± 0.000	1.879 ± 0.000	2.008 ± 0.000
DP-FTRL		1.589 ± 0.000	1.845 ± 0.000	1.867 ± 0.000	2.001 ± 0.000
PrE-Text		$\mathbf{1.456 \pm 0.003}$	$\mathbf{1.773 \pm 0.001}$	$\mathbf{1.771 \pm 0.000}$	$\mathbf{1.920 \pm 0.001}$
Expand-private	$\epsilon = \infty$	1.374 ± 0.002	1.682 ± 0.001	1.668 ± 0.001	1.585 ± 0.001

addition, PrE-Text uses 9× fewer rounds than the on-device baselines. This comes at the cost of more server-side compute in our experiments. However, using server resources is often more acceptable than using client resources.

4.2. Models stored on-server

We next consider the setting where models are stored and trained on-server (as opposed to on-device). This setting arises when the model is too large to fit on-device, for instance. On-device training and inference are infeasible in this setting. We instead obtain a DP synthetic dataset from the private federated training set and then finetune the server-side LLM on this synthetic dataset.

Experimental details. We use the same settings for PrE-text as in Section 4.1, except we only expand to 50000 samples (due to computational constraints of finetuning LLMs). We use LLaMA-2-7B as our evaluation model instead of DistilGPT2. We compare against the relevant and competitive baselines. For the non-finetuned (on private data) baseline, we simply report the evaluation loss on the private datasets for LLaMA-2-7B. To evaluate our proposed alternative, we finetune LLaMA-2-7B on the synthetic dataset produced by PrE-Text for one epoch with LoRA finetuning (with rank 4, $\alpha = 8$, applied to all the projection matrices in LLaMA-2-7B) with the AdamW optimizer at a learning rate of 0.0002 and a batch size of 512. For the Expand-private ($\epsilon = \infty$) baseline we finetune on 50000 samples expanded from the private train set. The on-device baselines are not appropriate for this setting. We also do not compare against DP-Prompt in this setting because it performed very poorly in Section 4.1, even worse than the $\epsilon = 0$ baseline.

Table 4. Next token prediction accuracy on private data for LLaMA-2-7B finetuned on PrE-Text synthetic data (under $(7.58, 3 \times 10^{-6})$-DP and $(1.29, 3 \times 10^{-6})$-DP). The finetuned model outperforms non-finetuned LLaMA-2-7B performance across four datasets. The error bars are stderr.

Method	Privacy	JOBS (↑)	FORUMS(↑)	MICROBLOG (↑)	CODE (↑)
Non-finetuned	$\epsilon = 0$	0.510 ± 0.000	0.466 ± 0.000	0.473 ± 0.000	0.460 ± 0.000
PrE-Text	$\epsilon = 1.29$	0.522 ± 0.000	0.479 ± 0.001	0.484 ± 0.000	0.473 ± 0.001
PrE-Text	$\epsilon = 7.58$	0.523 ± 0.000	0.480 ± 0.000	0.483 ± 0.000	0.473 ± 0.000
Expand-private	$\epsilon = \infty$	0.526 ± 0.000	0.484 ± 0.000	0.488 ± 0.001	0.476 ± 0.000

Table 5. Next token prediction cross-entropy loss on private data for LLaMA-2-7B finetuned on PrE-Text synthetic data (under $(7.58, 3 \times 10^{-6})$-DP and $(1.29, 3 \times 10^{-6})$-DP). The finetuned model outperforms non-finetuned LLaMA-2-7B performance across four datasets. The error bars are stderr.

Method	Privacy	JOBS (↓)	FORUMS(↓)	MICROBLOG (↓)	CODE (↓)
Non-finetuned	$\epsilon = 0$	2.915 ± 0.000	3.250 ± 0.000	3.220 ± 0.000	3.218 ± 0.000
PrE-Text	$\epsilon = 1.29$	2.811 ± 0.000	3.177 ± 0.004	3.133 ± 0.002	3.110 ± 0.000
PrE-Text	$\epsilon = 7.58$	2.795 ± 0.008	3.164 ± 0.005	3.139 ± 0.008	3.101 ± 0.000
Expand-private	$\epsilon = \infty$	2.746 ± 0.002	3.085 ± 0.001	3.088 ± 0.001	3.026 ± 0.000

Results. In Table 4 and Table 5 we observe that LLaMA-2-7B finetuned on PrE-Text DP synthetic data outperforms zero-shot LLaMA-2-7B. To the best of our knowledge, we are the first to demonstrate a way to privately finetune a large language model (that cannot fit on-device) in the federated setting. With major LLM providers running out of useful *public* data (Xu, 2022; Seetharaman, 2024; Needleman, 2024; Tong et al., 2024) for training, our proof-of-concept shows a promising new path forward: using *private* client data in a privacy-compliant and resource-conscious way.

4.3. Performance Scaling by Dataset Size

In this subsection, we investigate how the performance of PrE-Text scales as we change the number of samples generated in `Expand`. We finetune DistilGPT2 on DP synthetic datasets of sizes ranging from 50000 to 2 million, at $\epsilon = 7.58$ across the four private datasets JOBS, FORUMS, MICROBLOG, CODE. All the experimental details are the same as in the on-device PrE-Text experiment, except we scale the batch size according to the fraction of synthetic data we use over the full amount (2 million). For example, at a synthetic data size of 50000, we use [0.0025 $\times$ the original batch size] as the batch size. As shown in Figure 2, we find that like in (Honovich et al., 2022), the quality of the downstream model mostly improves log-linearly with the amount of synthetic data. However, in the case of CODE, the performance of the downstream model starts getting diminishing returns at around 1M samples, and even decreases in performance at 2M samples. The best number of synthetic samples for downstream may depend on the dataset.

5. Related Work

DP Federated Learning. Differentially Private Federated Learning (FL) is a widely-used approach for learning ML models from distributed private data (McMahan et al., 2017b; Kairouz et al., 2021c). In DP-FL, model weights are sent to users who train the model locally on their private data; private local model updates are then collected at a central server. Researchers have many techniques for improving privacy-utility tradeoffs in DP-FL. Some include shuffle-based privacy amplification (Bonawitz et al., 2016; Girgis et al., 2021; Agarwal et al., 2018), pretraining on public datasets (Xu et al., 2023c), private selection of the best pretraining datasets (Hou et al., 2023; Gu et al., 2022), and DP-FL methods that do not rely on uniform sampling/shuffling (Kairouz et al., 2021b).

Today, growing efforts study how to train larger models on client data. Charles et al. (2022) propose to have users optimize slices of large models, though they have not demonstrated the approach on models larger than shallow logistic regression and convolutional neural network models. Collins et al. (2023); Cai et al. (2022); Zhang et al. (2023a); Zhao et al. (2022a); Guo et al. (2023) only tune sub-components of the models in the federated setting to reduce client computational burden and communication, but this still requires having clients store and perform inference with large models on their device. Zhang et al. (2023c) use foundation models to produce synthetic data to pretrain smaller models. None of these methods consider privacy.

Synthetic Data. Synthetic data is increasingly being used to train language models (Taori et al., 2023; Wang et al., 2022; Honovich et al., 2022). Common approaches involve

Method	Per-Round Communication (Number of floats transmitted)		Per-sample Runtime (s)	Number of rounds
	Upload	*Download*		
DP-FL	82M	82M	3	100
PrE-Text	2048	800K	0.5	11
Reduction	41,000×	100×	6×	9×

Table 6. Communication and runtime cost of PrE-Text vs DP-FL. We find that PrE-Text achieves at least a 100× reduction in per-round communication, a 6× reduction in per-round runtime, and a 9× reduction in the number of rounds in our experiments. These costs are the same for all datasets and values of ϵ in our evaluation, for private clients with 8 samples each. The number of synthetic samples sent to each client for PrE-Text is 2048.

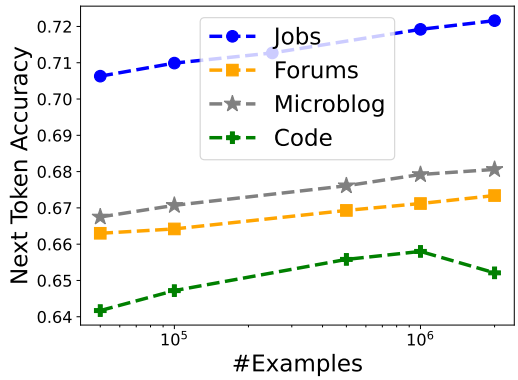

Figure 2. Next-token prediction accuracy for PrE-Text as we vary the number of synthetic examples generated by the `Expand` part of the algorithm. We find that increasing the number of synthetic examples across several orders of magnitude improves the accuracy of the downstream model (DistilGPT2) roughly log-linearly, though the growth peaks on CODE after 1M samples.

carefully designing prompts to ChatGPT (Radford et al., 2019) to generate synthetic training data for another open source model, e.g. LLaMA (Touvron et al., 2023), to replicate ChatGPT behavior. In these works, the synthetic data is used solely to enhance final model utility, and does not satisfy any formal privacy guarantees. In the image setting, useful synthetic data is often produced using dataset distillation (Wang et al., 2018; Zhao et al., 2020; Cazenavette et al., 2022). This approach has been adapted to the federated setting (Song et al., 2023).

DP Synthetic Data. Much of the work on producing *private* (i.e., DP) synthetic data from deep generative models is in the image domain (Lin et al., 2021; Cao et al., 2021; Dockhorn et al., 2022; Chen et al., 2022). Common approaches involve training a generative adversarial network (GAN) or diffusion model in a differentially private way, e.g., via DP-SGD, a DP version of stochastic gradient descent (SGD) (Abadi et al., 2016). Tang et al. (2023) generate DP synthetic text in the federated setting, but their method requires users to send private data to ChatGPT (which is located on-server). Such actions are not allowed under our threat model, where the central server is not trusted to hold private data. In concurrent work, (Xie et al., 2024) propose a private evolution algorithm for DP synthetic text data. Their method is similar to ours (we summarize differences in Section 1); the most important difference is that our focus is on understanding the relation between synthetic data and on device training, while their work aims to improve the quality of DP synthetic text data more generally. Also in the text domain, (Li et al., 2021; Yu et al., 2021a) demonstrate that it is possible to finetune pretrained large language models in a central-DP manner (i.e., the private data is available to the model developer).

Recently, several papers (some of them concurrent to ours) have proposed to finetune LLMs on DP synthetic data in the central DP setting (Yue et al., 2022; Kurakin et al., 2023; Yu et al., 2024; Ding et al., 2024). In the private federated setting, work by Wang et al. (2023a); Wu et al. (2024) also considers using LLM synthetic data. They propose to filter LLM synthetic data to samples that are relevant to the private data. The model is subsequently finetuned using DP-FL on-device. Their work shows a substantial improvement in next-word prediction accuracy compared to vanilla pre-training with DP-FL finetuning. In contrast, our work uses DP synthetic data to replace the DP-FL step itself, while keeping pre-training untouched. These methods are complementary and could potentially be combined. Overall, the literature suggests that DP synthetic data may be useful for training models from private, client data.

6. Conclusion

We propose PrE-Text, a method for generating privacy-preserving synthetic text from user devices that (1) surpasses federated learning approaches across privacy settings and (2) allow us to privately finetune LLMs on distributed user data. As model developers run out of useful data, our positive results suggest that DP synthetic data is a promising new privacy-preserving and resource-conscious data source for improving language models both big and small.

Impact Statement

PrE-Text relies on models that were pretrained on public data. It does not provide privacy guarantees for the public data that was used to train these models. Although that training data was public, it may not have been intended for use in language models, which raises questions about the ethics of using the resulting models (Tramèr et al., 2022). Therefore, in real deployments for PrE-Text, it is important to ensure that the public data used to pretrain the base models has been properly audited. Broadly, it is also important to communicate to users how much privacy protection they are getting when private algorithms are run on their data, and collect informed consent.

Acknowledgments

This work used Bridges-2 GPU at the Pittsburgh Supercomputing Center through allocation CIS240135 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, which is supported by National Science Foundation grants 2138259, 2138286, 2138307, 2137603, and 2138296 (Boerner et al., 2023). The authors acknowledge the National Artificial Intelligence Research Resource (NAIRR) Pilot and Delta GPU for contributing to this research result. GF and CH were supported in part by C3.ai, JP Morgan Chase, Bosch, Intel, and the Sloan Foundation.

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 308–318, 2016.

Abourayya, A., Kleesiek, J., Rao, K., Ayday, E., Rao, B., Webb, G., and Kamp, M. Little is enough: Improving privacy by sharing labels in federated semi-supervised learning. 2023. URL https://api.semanticscholar.org/CorpusID:263829949.

Agarwal, N., Suresh, A. T., Yu, F. X. X., Kumar, S., and McMahan, B. cpsgd: Communication-efficient and differentially-private distributed sgd. *Advances in Neural Information Processing Systems*, 31, 2018.

Agarwal, N., Kairouz, P., and Liu, Z. The skellam mechanism for differentially private federated learning. *Advances in Neural Information Processing Systems*, 34: 5052–5064, 2021.

Anil, R., Ghazi, B., Gupta, V., Kumar, R., and Manurangsi, P. Large-scale differentially private bert. *arXiv preprint arXiv:2108.01624*, 2021.

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.

Bagdasaryan, E., Kairouz, P., Mellem, S., Gascón, A., Bonawitz, K., Estrin, D., and Gruteser, M. Towards sparse federated analytics: Location heatmaps under distributed differential privacy with secure aggregation. *arXiv preprint arXiv:2111.02356*, 2021.

Bistritz, I., Mann, A., and Bambos, N. Distributed distillation for on-device learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 22593–22604. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/fef6f971605336724b5e6c0c12dc2534-Paper.pdf.

Boerner, T. J., Deems, S., Furlani, T. R., Knuth, S. L., and Towns, J. Access: Advancing innovation: Nsf's advanced cyberinfrastructure coordination ecosystem: Services & support. In *Practice and Experience in Advanced Research Computing*, pp. 173–176. 2023.

Bonawitz, K. A., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A., and Seth, K. Practical secure aggregation for federated learning on user-held data. In *NIPS Workshop on Private Multi-Party Machine Learning*, 2016. URL https://arxiv.org/abs/1611.04482.

Brown, H., Lee, K., Mireshghallah, F., Shokri, R., and Tramèr, F. What does it mean for a language model to preserve privacy? In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pp. 2280–2292, 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*, pp. 3192–3218. PMLR, 2023.

Bu, Z., Wang, Y.-X., Zha, S., and Karypis, G. Automatic clipping: Differentially private deep learning made easier and stronger. *Advances in Neural Information Processing Systems*, 36, 2024.

Cai, D., Wu, Y., Wang, S., Lin, F. X., and Xu, M. Fedadapter: Efficient federated learning for modern nlp. *arXiv preprint arXiv:2205.10162*, 2022.

Cao, T., Bie, A., Vahdat, A., Fidler, S., and Kreis, K. Don't generate me: Training differentially private generative models with sinkhorn divergence. *Advances in Neural Information Processing Systems*, 34:12480–12492, 2021.

Cazenavette, G., Wang, T., Torralba, A., Efros, A. A., and Zhu, J.-Y. Dataset distillation by matching training trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4750–4759, 2022.

Charles, Z., Bonawitz, K., Chiknavaryan, S., McMahan, B., et al. Federated select: A primitive for communication- and memory-efficient federated learning. *arXiv preprint arXiv:2208.09432*, 2022.

Charles, Z., Mitchell, N., Pillutla, K., Reneer, M., and Garrett, Z. Towards federated foundation models: Scalable dataset pipelines for group-structured learning. *arXiv preprint arXiv:2307.09619*, 2023.

Chen, J. and Yang, D. Unlearn what you want to forget: Efficient unlearning for llms. *arXiv preprint arXiv:2310.20150*, 2023.

Chen, J., Xu, W., Guo, S., Wang, J., Zhang, J., and Wang, H. Fedtune: A deep dive into efficient federated fine-tuning with pre-trained transformers. *arXiv preprint arXiv:2211.08025*, 2022.

Cho, Y. J., Wang, J., Chiruvolu, T., and Joshi, G. Personalized federated learning for heterogeneous clients with clustered knowledge transfer. *ArXiv*, abs/2109.08119, 2021. URL https://api.semanticscholar.org/CorpusID:237532588.

Collins, L., Wu, S., Oh, S., and Sim, K. C. Profit: Benchmarking personalization and robustness trade-off in federated prompt tuning. *arXiv preprint arXiv:2310.04627*, 2023.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Ding, Y., Wu, X., Meng, Y., Luo, Y., Wang, H., and Pan, W. Delving into differentially private transformer. *arXiv preprint arXiv:2405.18194*, 2024.

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. *Advances in Neural Information Processing Systems*, 36, 2024.

Dwork, C. Differential privacy. In *International colloquium on automata, languages, and programming*, pp. 1–12. Springer, 2006.

Dwork, C., Talwar, K., Thakurta, A., and Zhang, L. Analyze gauss: optimal bounds for privacy-preserving principal component analysis. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pp. 11–20, 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*, pp. 10611–10627. PMLR, 2023.

Girgis, A. M., Data, D., Diggavi, S., Kairouz, P., and Suresh, A. T. Shuffled model of federated learning: Privacy, accuracy and communication trade-offs. *IEEE journal on selected areas in information theory*, 2(1):464–478, 2021.

Grudzień, M., Malinovsky, G., and Richtárik, P. Can 5th generation local training methods support client sampling? yes! In *International Conference on Artificial Intelligence and Statistics*, pp. 1055–1092. PMLR, 2023.

Gu, X., Kamath, G., and Wu, S. Choosing public datasets for private machine learning via gradient subspace distance. In *NeurIPS 2022 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2022.

Gu, X., Kamath, G., and Wu, Z. S. Choosing public datasets for private machine learning via gradient subspace distance. *arXiv preprint arXiv:2303.01256*, 2023.

Gunasekar, S., Zhang, Y., Aneja, J., Mendes, C. C. T., Del Giorno, A., Gopi, S., Javaheripi, M., Kauffmann, P., de Rosa, G., Saarikivi, O., et al. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*, 2023.

Guo, T., Guo, S., Wang, J., Tang, X., and Xu, W. Promptfl: Let federated participants cooperatively learn prompts instead of models-federated learning in age of foundation model. *IEEE Transactions on Mobile Computing*, 2023.

Gupta, S., Huang, Y., Zhong, Z., Gao, T., Li, K., and Chen, D. Recovering private text in federated learning of language models. *Advances in neural information processing systems*, 35:8130–8143, 2022.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Honovich, O., Scialom, T., Levy, O., and Schick, T. Unnatural instructions: Tuning language models with (almost) no human labor. *arXiv preprint arXiv:2212.09689*, 2022.

Hou, C., Thekumparampil, K. K., Fanti, G., and Oh, S. Fedchain: Chained algorithms for near-optimal communication cost in federated learning. *arXiv preprint arXiv:2108.06869*, 2021a.

Hou, C., Thekumparampil, K. K., Fanti, G. C., and Oh, S. Efficient algorithms for federated saddle point optimization. *ArXiv*, abs/2102.06333, 2021b. URL https://api.semanticscholar.org/CorpusID:231918741.

Hou, C., Zhan, H., Shrivastava, A., Wang, S., Livshits, S., Fanti, G., and Lazar, D. Privately customizing prefinetuning to better match user data in federated learning. *arXiv preprint arXiv:2302.09042*, 2023.

Jang, J., Yoon, D., Yang, S., Cha, S., Lee, M., Logeswaran, L., and Seo, M. Knowledge unlearning for mitigating privacy risks in language models. *arXiv preprint arXiv:2210.01504*, 2022.

Kairouz, P., Ribero, M., Rush, K., and Thakurta, A. Dimension independence in unconstrained private erm via adaptive preconditioning. *ArXiv*, abs/2008.06570, 2020. URL https://api.semanticscholar.org/CorpusID:221140143.

Kairouz, P., Liu, Z., and Steinke, T. The distributed discrete gaussian mechanism for federated learning with secure aggregation. In *International Conference on Machine Learning*, pp. 5201–5212. PMLR, 2021a.

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, 2021b.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021c.

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S., and Suresh, A. T. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*, pp. 5132–5143. PMLR, 2020.

Kerrigan, G., Slack, D., and Tuyls, J. Differentially private language models benefit from public pre-training. *arXiv preprint arXiv:2009.05886*, 2020.

Kong, W. and Munoz Medina, A. A unified fast gradient clipping framework for dp-sgd. *Advances in Neural Information Processing Systems*, 36, 2024.

Kumar, V. B., Gangadharaiah, R., and Roth, D. Privacy adhering machine un-learning in nlp. *arXiv preprint arXiv:2212.09573*, 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.

Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J. E., Zhang, H., and Stoica, I. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461, 2019. URL http://arxiv.org/abs/1910.13461.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

Li, X., Tramer, F., Liang, P., and Hashimoto, T. Large language models can be strong differentially private learners. *arXiv preprint arXiv:2110.05679*, 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*, 35: 28616–28630, 2022.

Li, Y., Tan, Z., and Liu, Y. Privacy-preserving prompt tuning for large language model services. *arXiv preprint arXiv:2305.06212*, 2023.

Lin, T., Kong, L., Stich, S. U., and Jaggi, M. Ensemble distillation for robust model fusion in federated learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 2351–2363. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/18df51b97ccd68128e994804f3eccc87-Paper.pdf.

Lin, Z., Sekar, V., and Fanti, G. On the privacy properties of gan-generated samples. In *International Conference on Artificial Intelligence and Statistics*, pp. 1522–1530. PMLR, 2021.

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, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019. URL http://arxiv.org/abs/1907.11692.

Liu, Y., Suresh, A. T., Zhu, W., Kairouz, P., and Gruteser, M. Algorithms for bounding contribution for histogram estimation under user-level privacy. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 21969–21996. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/liu23ae.html.

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.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and Arcas, B. A. y. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Singh, A. and Zhu, J. (eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282. PMLR, 20–22 Apr 2017a. URL https://proceedings.mlr.press/v54/mcmahan17a.html.

McMahan, H. B., Ramage, D., Talwar, K., and Zhang, L. Learning differentially private recurrent language models. *arXiv preprint arXiv:1710.06963*, 2017b.

Mireshghallah, F., Inan, H. A., Hasegawa, M., Rühle, V., Berg-Kirkpatrick, T., and Sim, R. Privacy regularization: Joint privacy-utility optimization in language models. *CoRR*, abs/2103.07567, 2021. URL https://arxiv.org/abs/2103.07567.

Mishchenko, K., Malinovsky, G., Stich, S., and Richtárik, P. Proxskip: Yes! local gradient steps provably lead to communication acceleration! finally! In *International Conference on Machine Learning*, pp. 15750–15769. PMLR, 2022.

Nasr, M., Mahloujifar, S., Tang, X., Mittal, P., and Houmansadr, A. Effectively using public data in privacy preserving machine learning. In *International Conference on Machine Learning*, pp. 25718–25732. PMLR, 2023.

Needleman, S. Reddit to give openai access to its data in licensing deal. https://www.wsj.com/tech/ai/reddit-signs-data-licensing-deal-with-openai-14993757, 2024.

Neunhoeffer, M., Wu, Z. S., and Dwork, C. Private post-gan boosting. *ArXiv*, abs/2007.11934, 2020. URL https://api.semanticscholar.org/CorpusID:220713471.

Nguyen, J., Malik, K., Zhan, H., Yousefpour, A., Rabbat, M., Malek, M., and Huba, D. Federated learning with buffered asynchronous aggregation. In *International Conference on Artificial Intelligence and Statistics*, pp. 3581–3607. PMLR, 2022.

OpenAI. Openai. https://openai.com/policies/terms-of-use/, 2024. Accessed: 2024-05-24.

Papernot, N., Abadi, M., Erlingsson, Ú., Goodfellow, I. J., and Talwar, K. Semi-supervised knowledge transfer for deep learning from private training data. *ArXiv*, abs/1610.05755, 2016. URL https://api.semanticscholar.org/CorpusID:8696462.

Papernot, N., Song, S., Mironov, I., Raghunathan, A., Talwar, K., and Erlingsson, Ú. Scalable private learning with pate. *ArXiv*, abs/1802.08908, 2018. URL https://api.semanticscholar.org/CorpusID:3544583.

Ponomareva, N., Hazimeh, H., Kurakin, A., Xu, Z., Denison, C. E., McMahan, H. B., Vassilvitskii, S., Chien, S., and Thakurta, A. How to dp-fy ml: A practical guide to machine learning with differential privacy. *ArXiv*, abs/2303.00654, 2023. URL https://api.semanticscholar.org/CorpusID:257255451.

PyTorch. PyTorch Mobile, 2024. URL https://pytorch.org/mobile/home/.

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

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*, 2019.

Ramaswamy, S., Thakkar, O., Mathews, R., Andrew, G., McMahan, H. B., and Beaufays, F. Training production language models without memorizing user data. *arXiv preprint arXiv:2009.10031*, 2020.

Reimers, N. and Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL https://arxiv.org/abs/1908.10084.

Rozière, B., Gehring, J., Gloeckle, F., Sootla, S., Gat, I., Tan, X. E., Adi, Y., Liu, J., Remez, T., Rapin, J., et al. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*, 2023.

Sadiev, A., Kovalev, D., and Richtárik, P. Communication acceleration of local gradient methods via an accelerated primal-dual algorithm with an inexact prox. *Advances in Neural Information Processing Systems*, 35:21777–21791, 2022.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

Seetharaman, D. For data-guzzling ai companies, the internet is too small. https://www.wsj.com/tech/ai/ai-training-data-synthetic-openai-anthropic-9230f8d8, 2024.

Song, R., Liu, D., Chen, D. Z., Festag, A., Trinitis, C., Schulz, M., and Knoll, A. Federated learning via decentralized dataset distillation in resource-constrained edge environments. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–10. IEEE, 2023.

speedtest.net. United states median country speeds october 2023. https://www.speedtest.net/global-index/united-states, 2023. Accessed: 2023-11-27.

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*, 2023.

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*, 2024.

Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., and Hashimoto, T. B. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

Tensorflow. TensorFlow Federated, December 2018. URL https://github.com/tensorflow/federated.

Tong, A., Wang, E., and Coulter, M. Exclusive: Reddit in ai content licensing deal with google. https://www.reuters.com/technology/reddit-ai-content-licensing-deal-with-google-sources-say-2024-02-22/, 2024.

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 (CVPR) Workshops*, June 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*, 2023.

Tramèr, F. and Boneh, D. Differentially private learning needs better features (or much more data). *ArXiv*, abs/2011.11660, 2020. URL https://api.semanticscholar.org/CorpusID:227152280.

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

Utpala, S., Hooker, S., and Chen, P.-Y. Locally differentially private document generation using zero shot prompting. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 8442–8457, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.566. URL https://aclanthology.org/2023.findings-emnlp.566.

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., Li, Y., Xu, W., Li, R., Zhan, Y., and Zeng, Z. Dafkd: Domain-aware federated knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20412–20421, June 2023b.

Wang, J., Liu, Q., Liang, H., Joshi, G., and Poor, H. V. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 33:7611–7623, 2020.

Wang, J., Charles, Z., Xu, Z., Joshi, G., McMahan, H. B., Al-Shedivat, M., Andrew, G., Avestimehr, S., Daly, K., Data, D., et al. A field guide to federated optimization. *arXiv preprint arXiv:2107.06917*, 2021.

Wang, L., Zou, D., Patel, K. K., Wu, J., and Srebro, N. Private overparameterized linear regression without suffering in high dimensions. 2023c.

Wang, T., Zhu, J.-Y., Torralba, A., and Efros, A. A. Dataset distillation. *arXiv preprint arXiv:1811.10959*, 2018.

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.

Weller, O., Marone, M., Braverman, V., Lawrie, D., and Van Durme, B. Pretrained models for multilingual federated learning. *arXiv preprint arXiv:2206.02291*, 2022.

Wu, C., Wu, F., Lyu, L., Huang, Y., and Xie, X. Communication-efficient federated learning via knowledge distillation. *Nature Communications*, 13, 2021. URL https://api.semanticscholar.org/CorpusID:237353469.

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

Wu, T., Panda, A., Wang, J. T., and Mittal, P. Privacy-preserving in-context learning for large language models. In *The Twelfth International Conference on Learning Representations*, 2023.

Xie, C., Lin, Z., Backurs, A., Gopi, S., Yu, D., Inan, H. A., Nori, H., Jiang, H., Zhang, H., Lee, Y. T., et al. Differentially private synthetic data via foundation model apis 2: Text. *arXiv preprint arXiv:2403.01749*, 2024.

Xu, M., Song, C., Tian, Y., Agrawal, N., Granqvist, F., van Dalen, R., Zhang, X., Argueta, A., Han, S., Deng, Y., et al. Training large-vocabulary neural language models by private federated learning for resource-constrained devices. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023a.

Xu, T. We could run out of data to train ai language programs. https://www.technologyreview.com/2022/11/24/1063684/we-could-run-out-of-data-to-train-ai-language-programs/, 2022.

Xu, Z., Aggarwal, A., Feyisetan, O., and Teissier, N. On a utilitarian approach to privacy preserving text generation. *arXiv preprint arXiv:2104.11838*, 2021.

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, 2023b.

Xu, Z., Zhang, Y., Andrew, G., Choquette-Choo, C. A., Kairouz, P., McMahan, H. B., Rosenstock, J., and Zhang, Y. Federated learning of gboard language models with differential privacy. *arXiv preprint arXiv:2305.18465*, 2023c.

Yao, J., Chien, E., Du, M., Niu, X., Wang, T., Cheng, Z., and Yue, X. Machine unlearning of pre-trained large language models. *arXiv preprint arXiv:2402.15159*, 2024.

Yousefpour, A., Shilov, I., Sablayrolles, A., Testuggine, D., Prasad, K., Malek, M., Nguyen, J., Ghosh, S., Bharadwaj, A., Zhao, J., Cormode, G., and Mironov, I. Opacus: User-friendly differential privacy library in PyTorch. *arXiv preprint arXiv:2109.12298*, 2021.

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

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*, pp. 12208–12218. PMLR, 2021b.

Yu, D., Kamath, G., Kulkarni, J., Liu, T.-Y., Yin, J., and Zhang, H. Individual privacy accounting for differentially private stochastic gradient descent. *arXiv preprint arXiv:2206.02617*, 2022.

Yu, D., Kairouz, P., Oh, S., and Xu, Z. Privacy-preserving instructions for aligning large language models. *arXiv preprint arXiv:2402.13659*, 2024.

Yue, X., Inan, H. A., Li, X., Kumar, G., McAnallen, J., Shajari, H., 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, J., Vahidian, S., Kuo, M., Li, C., Zhang, R., Wang, G., and Chen, Y. Towards building the federated gpt: Federated instruction tuning. *arXiv preprint arXiv:2305.05644*, 2023a.

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

Zhang, T., Feng, T., Alam, S., Zhang, M., Narayanan, S. S., and Avestimehr, S. Gpt-fl: Generative pre-trained model-assisted federated learning. *arXiv preprint arXiv:2306.02210*, 2023c.

Zhao, B., Mopuri, K. R., and Bilen, H. Dataset condensation with gradient matching. *arXiv preprint arXiv:2006.05929*, 2020.

Zhao, H., Du, W., Li, F., Li, P., and Liu, G. Reduce communication costs and preserve privacy: Prompt

tuning method in federated learning. *arXiv preprint arXiv:2208.12268*, 2022a.

Zhao, J., Nguyen, J., and Shetty, S. Flsim. https://github.com/facebookresearch/FLSim, 2023.

Zhao, X., Li, L., and Wang, Y.-X. Provably confidential language modelling. *arXiv preprint arXiv:2205.01863*, 2022b.

Zhou, Y., Wu, Z. S., and Banerjee, A. Bypassing the ambient dimension: Private sgd with gradient subspace identification. *ArXiv*, abs/2007.03813, 2020. URL https://api.semanticscholar.org/CorpusID:220404588.

A. Additional Related Work

Federated Distillation. To sidestep the problem of high communication overheads, federated distillation has been proposed as a promising approach recently (Bistritz et al., 2020; Abourayya et al., 2023; Wang et al., 2023b; Wu et al., 2021; Lin et al., 2020; Cho et al., 2021), which draw inspiration from the PATE line of work (Papernot et al., 2016; 2018). As far as we know, these federated distillation methods have not developed algorithms for the private federated language setting that we study. There have been works studying DP synthetic data in settings outside of text (Torkzadehmahani et al., 2019; Neunhoeffer et al., 2020).

Clipping strategies. Clipping is a critical component of DP training of deep learning models, as it limits the sensitivity of the output. Li et al. (2021); Bu et al. (2024; 2023); Kong & Munoz Medina (2024) study how to (1) improve the computational efficiency of clipping, and (2) make it more privacy-efficient.

Unlearning. Chen & Yang (2023); Jang et al. (2022); Kumar et al. (2022); Yao et al. (2024) take a machine unlearning approach to remove sensitive data from a model. This approach avoids having to retrain a non-private model using private methods from scratch.

Private inference of language models. Tang et al. (2023); Wu et al. (2023) show how to do in-context learning with differential privacy. Duan et al. (2024) show how to protect the privacy of data that is used in in-context learning. Du et al. (2023) show how to add differential privacy in the forward pass of an LLM. Mattern et al. (2022) show how to anonymize data release in language models using differential privacy.

Private dataset selection. Gu et al. (2023); Hou et al. (2023) show how to privately choose pretraining datasets. Zhou et al. (2020) use gradient subspaces calculated from public data to reduce the amount of noise needed for differential privacy guarantees.

DP finetuning. Li et al. (2023) investigate differentially private prompt tuning. Li et al. (2022) show conditions under which LLMs do not suffer much accuracy drop-off from using DPSGD (Abadi et al., 2016). Yu et al. (2021b) show that by using a LoRA-like reparameterization, large language models can perform differentially private training without losing much performance. Nasr et al. (2023) modify DPSGD to more effectively use public data. Zhang et al. (2023b); Tang et al. (2024) show how to do zeroth order optimization of language models with differential privacy. Yu et al. (2024) show how to do DP instruction tuning. Kairouz et al. (2020) show that DP adagrad can achieve better regret guarantees than plain DP SGD. Tramèr & Boneh (2020) develop strong simple baselines for DP learning of models. Wang et al. (2023c) studies how to analyze overparameterized private linear regression via a theoretical lens. Yu et al. (2022) show how to account for privacy on an individual example level in DPSGD. Anil et al. (2021) show how scale allows product DP finetuning of BERT. Xu et al. (2021); Mireshghallah et al. (2021) consider DP language models satisfying empirical notions of privacy.

(Private) federated learning. Hou et al. (2021b) study how to do saddle point optimization in the federated setting. Weller et al. (2022) study the multilingual language model learning problem in federated learning. Ramaswamy et al. (2020) train production next-token-prediction models with differential privacy. Xu et al. (2023a) study the private federated learning setting, modifying private on-device training to be more noise and communication efficient. Gupta et al. (2022) show an attack on how to recover text used to train federated text models (if differential privacy is not used). Liu et al. (2023) study how to obtain low-error estimates of the aggregated histogram across user histograms under differential privacy.

Public pretraining. Ganesh et al. (2023) perform a theoretical analysis to show why public pretraining is so critical to the deployment of DP language models. Kerrigan et al. (2020) show that public pretraining helps with producing DP language models.

Best practices in DP published work. Brown et al. (2022); Zhao et al. (2022b) studies the issue of LLM pretraining data often not explicitly designated for public use, undermining typical assumptions made about public data in DP language models. Ponomareva et al. (2023) describe best practices when publishing DP work.

B. Prompt

We provide the prompt used for `expand` here.

List of 6 diverse original text samples:

Original Text Sample 1
As you meet with employers this summer, get in touch with the team and find out how they plan to find the person that will build their organization.

Original Text Sample 2
You can also of course change culture across your organization to ensure your team members work as a unit, with each working together to accomplish the company goal.

Original Text Sample 3
We're here to tell you how too take a close look and view the journey you've made here, based on how you left the hero behind.

Original Text Sample 4
Risk points can be validated in two or more ways. Here are some procedures that can be used in decision-making:

Original Text Sample 5
...

Figure 3. The synthetic data generation prompt for `Expand`. The blue text after "Original Text Sample 4" is generated. We parse the generated text for the text between Original Text Sample 4 and Original Text Sample 5 and use that as a synthetic sample.

C. Dataset generation details

We use the train split of the c4 English (c4-en) dataset (Raffel et al., 2019). We start by producing three federated private datasets from c4-en: JOBS, FORUMS, and MICROBLOG. We illustrate the process for JOBS; the process is similar for the other two. First, we take the first 11,000 samples in the c4-en dataset that come from a jobs site. The private train set consists of 10,000 randomly chosen samples, and the private evaluation set consists of the remaining 1000 samples. The federated dataset is then constructed by splitting the 10,000 training samples into 1250 clients of 8 samples each, split uniformly at random. The same is done for FORUMS (from an online forum site) and MICROBLOG (from an online microblogging site).

We also evaluate our method on a question-and-answer dataset focused on coding and technical topics, which has text content partitioned by users. We construct 1250 users by making a federated dataset where each client has the comments associated with that user in the dataset. For each user, we cap the number of comments to 128. This forms the training dataset. The evaluation dataset is made up of the first 2000 samples of the next 100 users from the same dataset. We call this dataset CODE.

For the initial population used in Algorithm 1, we take random samples from c4-en that are not in the private training sets nor even from the same website sources represented in the private datasets.

D. Experimental details

D.1. Privacy Details

For our privacy estimate in our evaluation, we use OPACUS.ACCOUNTANTS.ANALYSIS.RDP to compute the privacy guarantee for PrE-Text. We input $T = 11$ steps, $q = 1.0$, and set the NOISE_MULTIPLIER to be the ratio of σ to the sensitivity (the max number of samples per client for PrE-Text), setting σ to the value that gets us the desired ϵ value).

D.2. Baselines

(1) $\epsilon = 0$ baselines: We evaluate c4-only and Expand-only. c4-only is a DistilGPT2 model finetuned on a subset of c4-en that was not in any of the private datasets. As Xu et al. (2023c) found, finetuning on c4-en improved privacy-utility tradeoff greatly in next-token prediction, so this is an important baseline. Expand-only is a DistilGPT2 model finetuned on the subset of c4-en used in the c4-only baseline expanded to 2 million samples using `expand`. We use the AdamW optimizer with

a learning rate of 0.0002 and a batch size of 256 for c4-only, and a batch size of 65536 for Expand-only. We train for 20 epochs. The subset of c4-en is a subset of roughly 87k samples from c4-en that was not in any of the private datasets nor even from the same websites as any of the private datasets. We provide the precise dataset under data/initialization.json in the code repository.

(2) $\epsilon = \infty$ baselines: We evaluate Expand-private, which is a DistilGPT2 model finetuned on (1) the subset of c4-en used in c4-only baseline, and (2) the private dataset expanded to 2 million samples using `expand`. We found that this performed better than a model finetuned on only the private dataset itself. We use the AdamW optimizer with a learning rate of 0.0002 and a batch size of 256 for c4-only, and a batch size of 65536 and train for 20 epochs.

(3) On-device baselines: We evaluate DP-FedAvg (McMahan et al., 2017b) and DP-FTRL (Kairouz et al., 2021b) at privacy levels of $\epsilon = 1.29, \epsilon = 7.58$. We first finetune DistilGPT2 on the subset of c4-en used in the c4-only baseline, and then finetune it further using DP-FedAvg or DP-FTRL (which are on-device training methods). We use the DP-FTRL-TreeRestart variant of DP-FTRL, which makes the most sense in our setting which considers full participation in each communication round. For both DP-FedAvg and DP-FTRL, we tune the client learning rate in $\{0.1, 0.01, 0.001\}$, the number of communication rounds in $\{10, 20, 100\}$, and clipping in $\{1.0, 0.1, 0.01, 0.001\}$. Similar to (Kairouz et al., 2021b), we found that more rounds always helped, (i.e. 100 rounds is better than 20 and 10). The batch size is 4, and server momentum is 0.9. We report the best evaluation metric, evaluated at the end of each round.

(4) Text-to-text privatization baseline: In this approach, clients hold an LLM on-device (which may not be practical) and release privacy-preserved paraphrases of their text directly to the server. The representative method we use here is DP-Prompt (Utpala et al., 2023). We use the same prompt and model (flan-t5-3b) as Utpala et al. (2023). Note that these methods cannot take advantage of secure aggregation (text cannot be summed together) which necessitates much more noise to be added to the privatized text. We first finetune a DistilGPT2 model on the subset of c4-en used in the c4-only baseline, and then finetune it further on the privatized text received by the server. We clip according to the top and bottom logits just like Utpala et al. (2023). For the finetuning step, we use the AdamW optimizer with a learning rate of 0.0002 and a batch size of 256 and choose the best evaluation metric over 20 epochs.

(5) PrE-Text: We use PrE-Text to generate a synthetic dataset of 2 million samples. We first finetune DistilGPT2 on the subset of c4-en used in the c4-only baseline, and then finetune it further on the synthetic dataset generated by PrE-Text. The Φ we use is miniLM-L6-v2 (Reimers & Gurevych, 2019), which produces an embedding of size 384. In `Variation` we use RoBERTa-large as the mask filling model with top_p parameter set to 1.0 and temperature set to 1.0, $W_{\text{steps}} = 2$, and $\text{MASK}\% = 30\%$. In `Expand` we use LLaMA-2-7B with top_p set to 1.0 and temperature set to 1.0. When implementing `Expand` we use the library vLLM to speed up inference (Kwon et al., 2023). For $\epsilon = 1.29$: $N_{\text{syn}} = 1024$, $H = 5.9 \times 8.0 \times 1.541 \times \sqrt{2}$. For $\epsilon = 7.58$: $N_{\text{syn}} = 2048$, $H = 8.0 \times 1.541 \times \sqrt{2}$. For JOBS, FORUMS, MICROBLOG, there is a max of 8 samples per client, which limits the sensitivity to 8. For CODE, we deliberately clip the number of samples per client to 16, which limits sensitivity to 16. In CODE, both the noise and the threshold H is doubled compared to the other datasets to adjust to the increased sensitivity.