PRIVATELY CUSTOMIZING PREFINETUNING TO BETTER MATCH USER DATA IN FEDERATED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

In Federated Learning (FL), accessing private client data incurs communication and privacy costs. As a result, FL deployments commonly *prefinetune* (Aghajanyan et al., 2021) pre-trained foundation models on a (large, possibly public) dataset that is held by the central server; they then *FL-finetune* the model on a private, federated dataset held by clients (Nguyen et al., 2022). Evaluating prefinetuning dataset quality reliably and privately is therefore of high importance. To this end, we propose FreD (Federated Private Fréchet Distance) — a *privately* computed distance between a prefinetuning dataset and federated datasets. Intuitively, it privately computes and compares a Fréchet distance between embeddings generated by a large language model on both the central (public) dataset and the federated private client data. To make this computation privacy-preserving, we use distributed, differentially-private mean and covariance estimators. We show empirically that FreD accurately predicts the best prefinetuning dataset at minimal privacy cost. Altogether, using FreD we demonstrate a proof-of-concept for a new approach in private FL training: (1) customize a prefinetuning dataset to better match user data (2) prefinetune (3) perform FL-finetuning.

1 INTRODUCTION

Federated Learning (FL) is a framework in which a central server learns a model from data that is distributed across a set of clients, without directly accessing that data (McMahan et al., 2017; Kairouz et al., 2021). One of the main motivations for FL is privacy: an early hope was that by not accessing client data directly, the central server would learn less about it, thereby protecting client privacy. However, this intuition can be broken under naive implementations of FL (Carlini et al., 2021; Shokri et al., 2017); to achieve meaningful privacy, one needs provably private training mechanisms, e.g., using differential privacy (DP) Dwork et al. (2006); Abadi et al. (2016).

Despite its privacy benefits, DP training of FL models incurs high utility costs. For example, in the widely-used DP stochastic gradient descent (DP-SGD) Abadi et al. (2016), to achieve reasonable privacy guarantees, models can only be trained for a limited number of rounds Abadi et al. (2016).

To get around this challenge *under high privacy requirements where the number of FL training rounds is scarce*, a common approach is to "prefinetune" FL models (Nguyen et al., 2022). That is, given a pretrained foundation model (e.g., BERT), finetune it centrally on a dataset that is either public or owned by the FL coordinator, without privacy. The resulting *prefinetuned* model is used to initialize the federated model, which is sent to all clients and trained with private optimization. Prefinetuning helps the finetuning require fewer training steps, thereby boosting privacy guarantees. In this paper we use the term 'FL-finetuning' to refer to finetuning on federated datasets.

While FL-finetuning is widely used today (Nguyen et al., 2022), a crucial factor for its success is the choice of prefinetuning dataset. For example, when training a large language model (LLM), one could prefinetune on a number of public datasets—e.g., Reddit (Caldas et al., 2018) or StackOverflow (Reddi et al., 2020). The efficacy of pre-training will ultimately depend on how closely the prefinetuning dataset represents the true, private data (Gu et al., 2022; Tramèr et al., 2022).

Although prefinetuning dataset selection is critical to the success of FL finetuning, we lack algorithms to methodically select prefinetuning datasets, particularly in the FL setting (i.e., distributed private dataset under privacy constraints).

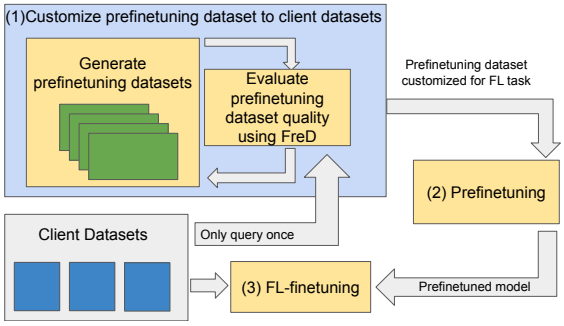

Figure 1: Proposed prefinetuning dataset customization approach using FreD to evaluate closeness of prefinetuning dataset to client data. We demonstrate that it is possible to privately and repeatedly evaluate prefinetuing dataset quality using FreD in step (1), and demonstrate the end-to-end approach experimentally.

Our work aims to fill this gap. Concurrent work (Gu et al., 2022) tackled a similar problem in the centralized setting; this method first computes two low-dimensional subspaces from the per-sample gradients for a small batch of public and private data, respectively. The dataset distance is measured as a projection metric between the two subspaces Edelman et al. (1998). The algorithm in (Gu et al., 2022) is designed for the centralized setting, and it is unclear how to apply it to a distributed dataset—particularly under the constraint that the distance measurement itself must be differentially private (Papernot & Steinke, 2021).

In this work, we instead modify a well-known dataset distance metric: the Fréchet Distance, most commonly used to measure dataset distances between synthetic generative adversarial network (GAN) and real datasets (Heusel et al., 2017) and occasionally used to measure distances between language datasets (Xiang et al., 2021). We demonstrate that a suitably chosen private formulation of the Fréchet Distance allows practitioners to accurately evaluate the quality of any number of prefinetuning datasets with respect to an FL task, with minimal privacy cost. Therefore, given a dataset generation mechanism that can take feedback from our private Fréchet Distance (this can be as simple as choosing between already existing datasets, or could be as sophisticated as using an LLM to generate datasets) one can effectively customize their prefinetuning dataset for the FL task. As a consequence, we make the first proof-of-concept for a new three-stage approach in privacy-sensitive FL applications Fig. 1: (1) customize the prefinetuning dataset to match client datasets (2) prefinetuning (3) FL-finetuning.

Contributions

- We present FreD, a privacy-preserving metric based on Fréchet Distance to measure dataset distance between prefinetuning and FL-finetuning datasets.[1] We show that FreD satisfies a formal (ϵ, δ)-differential privacy guarantee with respect to the private dataset. (Prop. 3.1)
- FreD computed with little privacy loss $(\epsilon = 0.6, \delta = 2 \times 10^{-6})$ empirically maintains enough resolution to accurately distinguish prefinetuning datasets that are only 1% different. (Fig. 2)
- We show that a smaller FreD between the prefinetuning dataset and the FL-finetuning dataset leads to better FL-finetuning performance, measured in terms of test perplexity (Table 2).
- Taken together, we are the first to privately customize prefinetuning to better match user data for better FL model performance. In our view, this is the main contribution of our work.

2 Preliminaries

We begin by defining differential privacy and introducing the Fréchet distance.

Definition 2.1 (Neighboring datasets)**.** Two datasets X, X' are said to be neighboring (denoted $X \sim X'$) if they differ in a single record. Note that we consider a *sample-level* notion of neighboring datasets; we allow one sample from one client to be added or removed.

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, we have

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

In this work, we will use a known DP mechanism called the Gaussian Mechanism Dwork et al. (2006), which adds Gaussian noise of a specific scale to protect privacy. To specify the scale, we must characterize the *sensitivity* of the statistical query we wish to release.

[1]Although FreD works well on non-federated settings, we focus on the federated setting in this work.

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 $\Delta g \triangleq \max_{X \sim X'} \|g(X) - g(X')\|_2$.

When X is a set of real-valued vectors (which is the case in our paper), bounding the ℓ_2 sensitivity of some g often requires an upper bound on the ℓ_2 norm of the vectors in X. The usual strategy to enforce this property for inputs to a DP mechanism is called *clipping*:

Definition 2.4 (Clipping)**.** We define the clipping operation $\chi_c : \mathbb{R}^d \to \mathbb{R}^d$ mapping a vector v to a clipped version with ℓ_2 norm at most c to be $\chi_c(v) \triangleq v/\text{max}(1, \frac{\|v\|_2}{c})$. Suppose E is some dataset of vectors of dimension d. We will overload notation by letting $\chi_c(E)$ be the dataset E where all its vectors have had the clipping operation χ_c performed on them.

DP Threat Model Our goal is to design DP algorithms in the high privacy regime (i.e. scarce FL training rounds) to defend against an adversary that accesses data at the central server. We assume the adversary has access to all intermediate quantities revealed to the central server. Our algorithm for computing FreD therefore relies in part on secure aggregation (Bonawitz et al., 2017), which allows the server to obtain a summary (e.g., sum) of the client data without access to individual client information. Because the central server only obtains a summary, the scale of noise required by the Gaussian Mechanism to achieve the same overall privacy is greatly reduced.

Fréchet distance The Fréchet distance is a distance metric over probability distributions. For two probability measures η and ν defined over $\mathbb{R}^n$, their Fréchet distance is defined as follows:

$$d(\eta, \nu) \triangleq \left(\inf_{\gamma \in \Gamma(\eta,\nu)} \int ||x - y||^2 d\gamma(x, y) \right)^{1/2},$$

where $\Gamma(\eta, \nu)$ is the set of all couplings of η and ν (i.e. the set of all distributions γ such that $\eta(x) = \int \gamma(x, y)dx$ and $\nu(y) = \int \gamma(x, y)dy$). In the special case where η and ν are Gaussian distributions $\mathcal{N}(\mu_1, \Sigma_1)$ and $\mathcal{N}(\mu_2, \Sigma_2)$, respectively, this can be written in closed form (Dowson & Landau, 1982):

$$d(\eta, \nu) = ||\mu_1 - \mu_2||_2^2 + \text{Tr}\left(\Sigma_1 + \Sigma_2 - 2\left(\Sigma_1\Sigma_2\right)^{1/2}\right). \tag{2}$$

Fréchet distance has been used in the GAN literature to evaluate the distance between synthetic and real datasets, using the Fréchet Inception Distance (FID) Heusel et al. (2017). The core idea of FID is to first extract representations of real and synthetic samples from the deepest hidden layer of a pre-trained Inception v3 model. Then, treating those representations as multivariate Gaussians, one estimates the empirical mean and covariance of each set. Finally, the distance between the two estimated distributions is computed using Fréchet distance (equation 2).

While FID (Heusel et al., 2017) is used for images, it also maintains semantic closeness in the case of language. Xiang et al. (2021) use BERT as the embedder and show that a smaller Fréchet distance corresponds to human notions of language dataset closeness. In this paper we use ALBERT, which is smaller and therefore better suited for FL. We adapt the Fréchet Distance to the space of training federated language models in FreD, where privacy constraints are critical.

3 FreD: Method

Let X_1 and X_2 denote language datasets of n_1 and n_2 sentences, respectively. Let $f : S \to \mathbb{R}^d$ be a sentence embedder that maps from the space S of sentences to a d-dimensional vector. We apply f to each sentence in X_1 and X_2 to produce $E_1 \in \mathbb{R}^{n_1 \times d}$ and $E_2 \in \mathbb{R}^{n_2 \times d}$. Then we calculate $m_1, m_2 \in \mathbb{R}^d$ to be the row-wise means of E_1, E_2 respectively and $C_1, C_2 \in \mathbb{R}^{d \times d}$ to be the row-wise covariances of E_1 and E_2 respectively.

The Frechét distance is then computed as in equation 2:

$$d(X_1, X_2) = \|m_1 - m_2\|_2^2 + \text{Tr}(C_1 + C_2 - 2(C_1C_2)^{1/2})$$

Now, let X_1 be the prefinetuning dataset and X_2 be the FL-finetuning dataset. Algorithm 1 describes how we compute the private FL version of the Fréchet Distance, FreD. At a high level, the algorithm's

Algorithm 1 FreD

Input: X_1, X_2 sentence datasets, where X_1 is the prefinetuning dataset (on server) and X_2 is the FL-finetuning dataset (distributed on clients), f sentence embedder, $\mathcal{C}$ client set, c embedding clipping norm

▷ **Compute mean and covariance of prefinetuning dataset**

Compute $E_1 = f(X_1)$, and $m_1 = \text{mean}(E_1)$, $C_1 = \text{Cov}(E_1)$

▷ **Compute DP mean of client datasets**

Let $\tau_1 = (2c/n_2)(\sqrt{2\log(1.25/\delta)}/\epsilon)$

Server sends f to all clients, clients compute E_2

Clients clip E_2 to get $\chi_c(E_2)$ and add $\mathcal{N}(0, \tau_1^2 I_{d\times d})$ to each sample in $\chi_c(E_2)$ for $\mathcal{M}_1(\chi_c(E_2))$

Server securely aggregates mean of $\mathcal{M}_1(\chi_c(E_2))$ from the clients to get pmean_2.

▷ **Compute DP covariance of client datasets**

Let $\tau_2 = (c^2/n_2)(\sqrt{2\log(1.25/\delta)}/\epsilon)$

Server sends pmean_2, clients subtract pmean_2 from each sample in $\chi_c(E_2)$ for $\Theta(\chi_c(E_2))$

Clients clip $\Theta(\chi_c(E_2))$ to get $\chi_c(\Theta(\chi_c(E_2)))$

Clients compute their contributions to $\tilde{C}_2 := (1/n_2)\chi_c(\Theta(\chi_c(E_2)))^\top \chi_c(\Theta(\chi_c(E_2)))$

Clients add $\mathcal{N}(0, \tau_2^2)$ independently to the upper triangle of their contributions of $\tilde{C}_2$, mirror the results to the lower triangle, and get $\mathcal{M}_2(\tilde{C}_2)$

Server securely aggregates $\mathcal{M}_2(\tilde{C}_2)$

Server projects $\mathcal{M}_2(\tilde{C}_2)$ to the nearest PSD matrix for pcov_2

▷ **Compute FreD**

Server computes $\|m_1 - \mathsf{pmean}_2\|_2^2 + \text{Tr}(C_1 + \mathsf{pcov}_2 - 2(C_1\mathsf{pcov}_2)^{1/2})$

core is as follows: (1) we first calculate pmean_2 from the clients, which is the DP mean from E_2 (2) we send the DP mean to clients who then center their embeddings using the DP mean (the embeddings will not be exactly zero mean because the DP mean is not the true mean, but we find this is sufficient), and then calculate the DP covariance from these centered embeddings (3) we use the DP mean and DP covariance of the client data together with the mean and covariance of the prefinetuning dataset to get FreD. We will now justify the scale of the noise we add in Algorithm 1.

Proposition 3.1. *Let $\tau_1 = (2c/n_2)(\sqrt{2\log(1.25/\delta)}/\epsilon)$ and $\tau_2 = (c^2/n_2)(\sqrt{2\log(1.25/\delta)}/\epsilon)$ be the scale of the Gaussian noise added in Algorithm 1. Then calculating FreD as in Algorithm 1 satisfies $(2\epsilon, 2\delta)$-DP.*

Proof. From Dwork et al. (2014)[Theorem 2], given that the ℓ_2 sensitivity of the mean of $\chi_c(E_2)$ is $2c/n_2$, we know that the Gaussian mechanism with noise on the scale of τ_1 maintains (ϵ, δ)-DP.

Next, again observe from Dwork et al. (2014)[Theorem 2] that for a vector dataset A (and its neighbor A', vectors arranged row-wise), the ℓ_2 sensitivity of $A^\top A'$ if each row has norm at most 1 is $\|A^\top A - A'^\top A'\|_2 \leq 1$. We can write $B := \chi_c(\Theta(\chi_c(E_2)))$ as cA for some A. Furthermore, we can also write B' (B' a neighbor of B) as cA' where A' is a neighbor of A. Therefore,

$$\|B^\top B - B'^\top B'\|_2 = \|(cA)^\top(cA) - (cA')^\top(cA')\|_2 = c^2\|A^\top A - A'^\top A'\|_2 \leq c^2. \quad (3)$$

Therefore, the Gaussian mechanism with noise on the scale of τ_2 maintains (ϵ, δ)-DP for the released private covariance $\mathcal{M}_2(\tilde{C}_2)$. By the sequential composition property of (ϵ, δ)-DP, releasing both the mean and the covariance in this way satisfies $(2\epsilon, 2\delta)$-DP. □

4 EXPERIMENTS

4.1 CHOOSING BETWEEN TWO PREFINETUNING DATASETS

In this subsection we study the case where a practitioner is choosing between two existing candidate prefinetuning datasets.

Experimental Setup In this subsection we use 3 datasets in our experiments: the StackOverflow language dataset (Reddi et al., 2020), the Reddit language dataset derived from Reddit data released

Prefinetune data	FL-finetune data	Non-private FreD	$(0.6, 10^{-6})$-DP FreD	Perplexity Reddit-test
StackOverflow-train	Reddit-train	678.12	826.82	65.90
Wikitext-train	Reddit-train	877.58	954.25	67.33

Table 1: The closer dataset to Reddit (StackOverflow) leads to better FL-finetuning performance on Reddit. Moveover, we can identify that StackOverflow is the closer dataset to Reddit even when computing FreD privately.

by `pushshift.io` (Caldas et al., 2018), and the Wikitext dataset (Merity et al., 2016). Here we let StackOverflow-train and Wikitext-train be the possible choices for prefinetuning datasets and Reddit-train be the FL-finetuning dataset. Performance is evaluated on Reddit-test. All three are freely available open-source. We use a DistilGPT-2 model (Sanh et al., 2019), initialized with weights prefinetuned on various combinations of our public datasets. The task is to use DistilGPT-2 to perform next word prediction. The metric we use for this task is perplexity (Jelinek et al., 1977). We train on the cross-entropy loss.

Training Details In the prefinetuning stage, the batch size is 16 and we tuned the best learning rate using a Bayesian hyperparameter sweep over the range $[10^{-1}, 10e^{-6}]$ on the SGD optimizer. We choose the representative prefinetuned model based on its performance on the validation set: i.e. if we train on StackOverflow-train, the choice is based on performance on StackOverflow-val. We prefinetune for 10 epochs. For the FL-finetuning stage, we select hyperparameters similarly given the prefinetuning initialization. We perform the FL-finetuning non-privately.

Results In Table 1, the prefinetuning dataset closest to Reddit, StackOverflow, performs the best as the prefinetuning dataset. Furthermore, when we calculate FreD under $(0.6, 10^{-6})$-DP, we can still easily identify which of Wikitext and StackOverflow is closer. This experiment demonstrates end-to-end our proposed method of prefinetuning dataset customization, at all three steps Fig. 1.

4.2 Choosing among a sequence of prefinetuning datasets

The motivation of this setting is that, because calculating FreD Algorithm 1 only requires us to query the private user data once (to calculate pmean_2 and pcov_2), we can generate a sequence of datasets (the dataset generation can be as basic as finding existing public datasets and as sophisticated as generating synthetic data from LLMs) and evaluate their suitability for the FL task for no additional privacy cost after the first FreD calculation. Here, we generate a sequence of prefinetuning datasets, which are a mix of the StackOverflow and Wikitext datasets, and we evaluate FL-finetuning performance on a split of StackOverflow. The goal here is to show that a highly private FreD metric (1) predicts the FL-finetuning performance with respect to prefinetuning dataset choice (2) can accurately tell apart prefinetuning datasets that are even very similar.

Experimental Setup In this subsection we use two datasets in our experiments: the StackOverflow language dataset (Reddi et al., 2020) and the Wikitext dataset (Merity et al., 2016). We split StackOverflow-train into two datasets of equal size: StackOverflow-train1 and StackOverflow-train2. We let our choices of prefinetuning dataset be $Y\%$ Stackoverflow-train1 and $100-Y\%$ Wikitext-train, where Y can vary between 0 and 100. The overall dataset size is kept constant at 150k sentences. When FL-finetuning, we FL-finetune on Stackoverflow-train2 (users and data are distributed). We test on Stackoverflow-test before and after finetuning. We use a DistilGPT-2 model (Sanh et al., 2019), initialized with weights prefinetuned on various combinations of our public datasets. The task is to use DistilGPT-2 to perform next word prediction. The metric we use for this task is perplexity (Jelinek et al., 1977). We train to minimize the cross-entropy loss.

Training Details In the prefinetuning stage, the batch size is 16 and used learning rate 0.002 with momentum 0.9, on the SGD optimizer. We prefinetune for 50 epochs. For FL-finetuning stage, we FL-finetune with noise scale of 1.5 and clipping 0.01, using the Opacus framework (Yousefpour

Y%	Test perplexity before FL-finetune	Test perplexity after FL-finetune
10	56.18	52.77
40	44.01	42.47
70	39.35	38.52
95	37.18	36.59
99	36.90	36.37
100	36.88	36.33

Table 2: Test perplexity before and after DP FL-finetuning, as a function of Y%, the percent of the prefinetuning dataset that is composed of StackOverflow-train1. We observe that as Y% increases, the better our model performs after prefinetuning (but before FL-finetuning) and also after FL-finetuning. We also observe that the improvement from doing FL-finetuning on top of prefinetuning decreases as Y% increases.

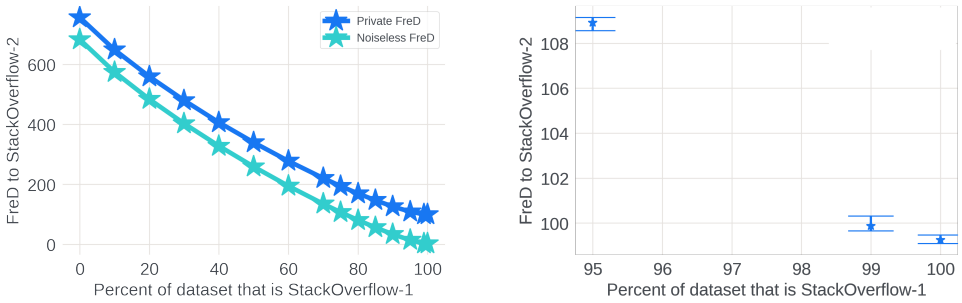

Figure 2: Left: We see that both private ($\epsilon = 0.6, \delta = 2 \times 10^{-6}$) and non-private FreD monotonically decrease with the percentage of prefinetuning dataset that is StackOverflow-train1. Right: We see that even between datasets that are 1% apart, private FreD over 5 trials has nonoverlapping error bars–i.e. the highest observed FreD value for 100% StackOverflow-train1 is still lower than the lowest observed FreD value for 99% StackOverflow-train1. This demonstrates that private FreD can distinguish between highly similar datasets with confidence.

et al., 2021). We train for one epoch, and sample 100 clients per round. The privacy cost incurred is ($\epsilon = 1.26, \delta = 10^{-6}$) from this stage.

Results First, in Table 2, we see that a closer dataset (as Y increases, the closer our prefinetuning dataset) the better the test perplexity before and after finetune. Furthermore, the gain we get from FL-finetuning over only prefinetuning decreases as Y increases. Next, we observe Fig. 2 that both non-private FreD and private ($\epsilon = 0.6, \delta = 2 \times 10^{-6}$) FreD corresponds strongly with Y, showing that even under strong privacy requirements, FreD still gives high-resolution information about the comparative closeness between two choices of prefinetuning datasets. In particular, in Fig. 2 (right), we see that evan for prefinetuning datasets that are only 1% apart, it is possible to distinguish then using ($\epsilon = 0.6, \delta = 2 \times 10^{-6}$) FreD with high confidence. This experiment shows another end-to-end example of our new proposed prefinetuning customization process Fig. 1, demonstrating that by using FreD, customization can be highly accurate.

5 Conclusion

In this paper, we make the case for using FreD as an informative metric for prefinetuning dataset choice. Our experiments show that FreD is a good indicator of the quality of the prefinetuning dataset for an FL task. By demonstrating this, we show that we can use FreD to privately customize our prefinetuning to match user data, which improves FL model performance. Altogether, we demonstrate the first proof-of-concept of a new approach in privacy-sensitive FL applications: customization of the prefinetuning dataset for better FL model performance. In the future, we plan to introduce more powerful prefinetuning dataset generation strategies to augment the power of our approach.

REFERENCES

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 308–318, 2016.

Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. Muppet: Massive multi-task representations with pre-finetuning. *arXiv preprint arXiv:2101.11038*, 2021.

Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, pp. 1175–1191, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450349468. doi: 10.1145/3133956.3133982. URL https://doi.org/10.1145/3133956.3133982.

Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 2633–2650, 2021.

DC Dowson and BV666017 Landau. The fréchet distance between multivariate normal distributions. *Journal of multivariate analysis*, 12(3):450–455, 1982.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin (eds.), *Theory of Cryptography*, pp. 265–284, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-32732-5.

Cynthia Dwork, Kunal Talwar, Abhradeep Thakurta, and Li Zhang. 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.

Alan Edelman, Tomás A Arias, and Steven T Smith. The geometry of algorithms with orthogonality constraints. *SIAM journal on Matrix Analysis and Applications*, 20(2):303–353, 1998.

Xin Gu, Gautam Kamath, and Steven Wu. Choosing public datasets for private machine learning via gradient subspace distance. In *NeurIPS 2022 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2022. URL https://openreview.net/forum?id=zr6AZ8ARan.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Fred Jelinek, Robert L Mercer, Lalit R Bahl, and James K Baker. Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1): S63–S63, 1977.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016.

John Nguyen, Jianyu Wang, Kshitiz Malik, Maziar Sanjabi, and Michael Rabbat. Where to begin? on the impact of pre-training and initialization in federated learning. *arXiv preprint arXiv:2210.08090*, 2022.

Nicolas Papernot and Thomas Steinke. Hyperparameter tuning with renyi differential privacy. *arXiv preprint arXiv:2110.03620*, 2021.

Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *NeurIPS EMC*$^2 Workshop$, 2019.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pp. 3–18. IEEE, 2017.

Florian Tramèr, Gautam Kamath, and Nicholas Carlini. Considerations for differentially private learning with large-scale public pretraining. *arXiv preprint arXiv:2212.06470*, 2022.

Jiannan Xiang, Yahui Liu, Deng Cai, Huayang Li, Defu Lian, and Lemao Liu. Assessing dialogue systems with distribution distances. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 2192–2198, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.193. URL https://aclanthology.org/2021.findings-acl.193.

Ashkan Yousefpour, Igor Shilov, Alexandre Sablayrolles, Davide Testuggine, Karthik Prasad, Mani Malek, John Nguyen, Sayan Ghosh, Akash Bharadwaj, Jessica Zhao, et al. Opacus: User-friendly differential privacy library in pytorch. *arXiv preprint arXiv:2109.12298*, 2021.