Practical and Private (Deep) Learning Without Sampling or Shuffling

Peter Kairouz [1] **Brendan McMahan** [1] **Shuang Song** [1] **Om Thakkar** [1] **Abhradeep Thakurta** [1] **Zheng Xu** [1]

Abstract

We consider training models with differential privacy (DP) using mini-batch gradients. The existing state-of-the-art, Differentially Private Stochastic Gradient Descent (DP-SGD), requires *privacy amplification by sampling or shuffling* to obtain the best privacy/accuracy/computation trade-offs. Unfortunately, the precise requirements on exact sampling and shuffling can be hard to obtain in important practical scenarios, particularly federated learning (FL). We design and analyze a DP variant of Follow-The-Regularized-Leader (DP-FTRL) that compares favorably (both theoretically and empirically) to amplified DP-SGD, while allowing for much more flexible data access patterns. DP-FTRL does not use any form of privacy amplification.

1. Introduction

Differentially private stochastic gradient descent (DP-SGD) (Song et al., 2013; Bassily et al., 2014; Abadi et al., 2016) has become state-of-the-art in training private (deep) learning models (Abadi et al., 2016; McMahan et al., 2018; Erlingsson et al., 2020; Papernot et al., 2020b; Facebook, 2020; Tramèr & Boneh, 2021). It operates by running stochastic gradient descent (Robbins & Monro, 1951) on noisy mini-batch gradients[1], with the noise calibrated such that it ensures differential privacy. The privacy analysis heavily uses tools like *privacy amplification by sampling/shuffling* (Kasiviswanathan et al., 2008; Bassily et al., 2014; Abadi et al., 2016; Wang et al., 2019; Zhu & Wang, 2019; Erlingsson et al., 2019; Feldman et al., 2020b) to obtain the best privacy/utility trade-offs. Such amplification tools require that each mini-batch is a perfectly (uniformly) random subset of the training data. This assumption can make practical deployment prohibitively hard, especially in the context of distributed settings like federated learning (FL) where one has little control on which subset of the training data one sees at any time (Kairouz et al., 2019; Balle et al., 2020).

We propose a new online learning (Hazan, 2019; Shalev-Shwartz et al., 2011) based DP algorithm, *differentially private follow-the-regularized-leader* (DP-FTRL), that has privacy/utility/computation trade-offs that are competitive with DP-SGD, and does not rely on privacy amplification. DP-FTRL *significantly outperforms* un-amplified DP-SGD at all privacy levels. In the higher-accuracy / lower-privacy regime, DP-FTRL outperforms even *amplified* DP-SGD. We emphasize that in the context of ML applications, using a DP mechanism even with a large ε is practically much better for privacy than using a non-DP mechanism (Song & Shmatikov, 2019; Jagielski et al., 2020; Thakkar et al., 2020; Nasr et al., 2021).

Privacy amplification and its perils: At a high-level, DP-SGD can be thought of as an iterative noisy state update procedure for T steps operating over mini-batches of the training data. For a time step $t \in [T]$ and an arbitrary mini-batch of size k from a data set D of size n, let σ_t be the standard deviation of the noise needed in the t^{th} update to satisfy ε_t-differential privacy. If the mini-batch is chosen *u.a.r. and i.i.d.* from D at each time step[2] t, then privacy amplification by sampling (Kasiviswanathan et al., 2008; Bassily et al., 2014; Abadi et al., 2016; Wang et al., 2019) allows one to scale down the noise to $\sigma_t \cdot (k/n)$, while still ensuring ε_t-differential privacy.[3] Such amplification is crucial for DP-SGD to obtain state-of-the-art models in practice (Abadi et al., 2016; Papernot et al., 2020b; Tramèr & Boneh, 2021) when $k \ll n$.

There are two major bottlenecks for such deployments: i) For large data sets, achieving uniform sampling/shuffling of the mini-batches in every round (or epoch) can be pro-

[1]Google. Correspondence to: Abhradeep Thakurta <athakurta@google.com>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

[1]Gradient computed on a subset of the training examples, also called a mini-batch.

[2]One can also create a mini-batch with Poisson sampling (Abadi et al., 2016; McMahan et al., 2017b; Zhu & Wang, 2019), except the batch size is now a random variable. For brevity, we focus on the fixed batch setting.

[3]A similar argument holds for amplification by shuffling (Erlingsson et al., 2019; Feldman et al., 2020b), when the data are uniformly shuffled at the beginning of every epoch.We do not consider privacy amplification by iteration (Feldman et al., 2018) in this paper, as it only applies to smooth convex functions.

hibitively expensive in terms of computation and/or engineering complexity, ii) In distributed settings like federated learning (FL) (McMahan et al., 2017a), uniform sampling/shuffling may be infeasible to achieve because of widely varying available population at each time step. Our work answers the following question in affirmative: *Can we design an algorithm that does not rely on privacy amplification, and hence allows data to be accessed in an arbitrary order, while providing privacy/utility/computation trade-offs competitive with DP-SGD?*

DP-FTRL and amplification-free model training: DP-FTRL can be viewed as a differentially private variant of the follow-the-regularized-leader (FTRL) algorithm (Xiao, 2010; McMahan, 2011; Duchi et al., 2011). The main idea in DP-FTRL is to use the *tree aggregation trick* (Dwork et al., 2010; Chan et al., 2011) to add noise to the sum of mini-batch gradients, in order to ensure privacy. Crucially, it deviates from DP-SGD by adding correlated noise across time steps, as opposed to independent noise. This particular aspect of DP-FTRL allows it to get strong privacy/utility trade-off without relying on privacy amplification.

Federated Learning (FL) and DP-FTRL: There has been prior work (Balle et al., 2020; Ramaswamy et al., 2020) detailing challenges for obtaining strong privacy guarantees that incorporate limited availability of participating clients in real-world applications of Federated Learning. Although there exist techniques like the Random Check-Ins (Balle et al., 2020) that obtain privacy amplification for FL settings, implementing such techniques may still require clients to keep track of the number of training rounds being completed at the server during their period(s) of availability to be able to uniformly randomize their participation. On the other hand, since the privacy guarantees of DP-FTRL (Algorithm 1) do not depend on any type of privacy amplification, it does not require any local/central randomness apart from noise addition to the model updates.

Appendices A and Section 2 describe additional related work and background, respectively.

1.1. Problem Formulation

Suppose we have a stream of data samples $D = [d_1, \ldots, d_n] \in \mathcal{D}^n$, where $\mathcal{D}$ is the domain of data samples, and a loss function $\ell : \mathcal{C} \times \mathcal{D} \to \mathbb{R}$, where $\mathcal{C} \in \mathbb{R}^p$ is the space of all models. We consider the following two problem settings.

Regret Minimization: At every time step $t \in [n]$, while observing samples $[d_1, \ldots, d_{t-1}]$, the algorithm $\mathcal{A}$ outputs a model $\theta_t \in \mathcal{C}$ which is used to predict on example d_t. The performance of $\mathcal{A}$ is measured in terms of regret against an arbitrary post-hoc comparator $\theta^* \in \mathcal{C}$:

$$R_D(\mathcal{A}; \theta^*) = \frac{1}{n}\sum_{t=1}^{n} \ell(\theta_t; d_t) - \frac{1}{n}\sum_{t=1}^{n} \ell(\theta^*; d_t). \quad (1)$$

We consider the algorithm $\mathcal{A}$ low-regret if $R_D(\mathcal{A}; \theta^*) = o(1)$. To ensure a low-regret algorithm, we will assume $\|\nabla \ell(\theta; d)\|_2 \leq L$ for any data sample d, and any models $\theta \in \mathcal{C}$. We consider both *adversarial regret*, where the data sample d_t are drawn adversarially based on the past output $\{\theta_1, \ldots, \theta_t\}$ (Hazan, 2019), and *stochastic regret* (Hazan & Kale, 2014), where the data samples in D are drawn i.i.d. from some fixed distribution τ.

Excess Risk Minimization: In this setting, we look at the problem of minimizing the excess population risk. Assuming the data set D is sampled i.i.d. from a distribution τ, and the algorithm $\mathcal{A}$ outputs $\widehat{\theta} \in \mathcal{C}$, we want to minimize

$$\mathsf{PopRisk}(\mathcal{A}) = \mathbb{E}_{d\sim\tau}\ell(\widehat{\theta}; d) - \min_{\theta\in\mathcal{C}} \mathbb{E}_{d\sim\tau}\ell(\theta; d). \quad (2)$$

All the algorithms in this paper guarantee differential privacy (Dwork et al., 2006b;a) and Rényi differential privacy (Mironov, 2017) (See Section 2 for details). The definition of a single data record can be one training example (a.k.a., *example level* privacy), or a group of training examples from one individual (a.k.a., *user level* privacy). Except for the empirical evaluations in the FL setting, we focus on example level privacy.

Definition 1.1 (Differential privacy (Dwork et al., 2006b;a))**.** *A randomized algorithm $\mathcal{A}$ is (ε, δ)-differentially private if for any neighboring data sets D, D' that differ in one record, and for any event $\mathcal{S}$ in the output range of $\mathcal{A}$, we have*

$$\mathbf{Pr}[\mathcal{A}(D) \in \mathcal{S}] \leq e^{\varepsilon} \cdot \mathbf{Pr}[\mathcal{A}(D') \in \mathcal{S}] + \delta,$$

where the probability is over the randomness of $\mathcal{A}$.

1.2. Our Contributions

Our primary contribution in this paper is a private online learning algorithm: differentially private follow-the-regularized leader (DP-FTRL) (Algorithm 1). We provide tighter privacy/utility trade-offs based on DP-FTRL (see Table 1 for a summary), and show how it can be easily adapted to train (federated) deep learning models, with comparable, and sometimes even better privacy/utility/computation trade-offs as DP-SGD. We summarize these contributions below.

DP-FTRL algorithm: We provide DP-FTRL, a differentially private variant of the Follow-the-regularized-leader (FTRL) algorithm (McMahan & Streeter, 2010; McMahan,

Table 1. Best known regret guarantees. Here, the high probability means w.p. at least $1-\beta$ over the the randomness of the algorithm. The expected regret is an expectation over the random choice of the data set and the randomness of the algorithm.

Class	Adversarial Regret		Stochastic Regret	
	Expected	High probability	Expected	High probability
Least-squares (and linear)	$O\left(\left(\frac{1}{\sqrt{n}}+\frac{\sqrt{p}}{\varepsilon n}\right)\cdot \mathsf{polylog}\left(\frac{1}{\delta}, n\right)\right)$ (Agarwal & Singh, 2017)	Same as general convex	$O\left(\left(\frac{1}{\sqrt{n}}+\frac{\sqrt{p}}{\varepsilon n}\right)\cdot \mathsf{polylog}\left(\frac{1}{\delta}, n\right)\right)$ (Agarwal & Singh, 2017)	$O\left(\left(\frac{1}{\sqrt{n}}+\frac{\sqrt{p}}{\varepsilon n}\right)\cdot \mathsf{polylog}\left(\frac{1}{\delta}, n, \frac{1}{\beta}\right)\right)$ [Theorem 4.3]
General convex	Constrained and unconstrained: $O\left(\left(\frac{1}{\sqrt{n}}+\frac{p^{1/4}}{\sqrt{\varepsilon n}}\right)\cdot \mathsf{polylog}\left(\frac{1}{\delta}, n, \frac{1}{\beta}\right)\right)$ [Theorem 4.1]			

2011; Shalev-Shwartz et al., 2011; Hazan, 2019) for online convex optimization (OCO). We also provide a variant called the momentum DP-FTRL that has superior performance in practice. (Agarwal & Singh, 2017) provided a instantiation of DP-FTRL specific to linear losses. (Smith & Thakurta, 2013) provided an algorithm similar to DP-FTRL, where instead of just linearizing the loss, a quadratic approximation to the regularized loss was used.

Regret guarantees: In the adversarial OCO setting (Section 4.1), compared to prior work (Jain et al., 2012; Smith & Thakurta, 2013; Agarwal & Singh, 2017), DP-FTRL has the following major advantages. First, it improves the best known regret guarantee in (Smith & Thakurta, 2013) by a factor of $\sqrt{\varepsilon}$ (from $\widetilde{O}\left(\sqrt{\frac{\sqrt{p}}{\varepsilon^2 n}}\right)$ to $\widetilde{O}\left(\sqrt{\frac{\sqrt{p}}{\varepsilon n}}\right)$, when $\varepsilon \leq 1$). This improvement is significant because it *distinguishes centrally private OCO from locally private* (Warner, 1965; Evfimievski et al., 2003; Kasiviswanathan et al., 2008) OCO[4]. Second, unlike (Smith & Thakurta, 2013), DP-FTRL (and its analysis) extends to the unconstrained setting $\mathcal{C} = \mathbb{R}^p$. Also, in the case of composite losses (Duchi et al., 2010; Xiao, 2010; McMahan, 2011; 2017), i.e., where the loss functions are of the form $\ell(\theta; d_t) + r_t(\theta)$ with $r : \mathcal{C} \to \mathbb{R}^+$ (e.g., $\|\cdot\|_1$) being a convex regularizer, DP-FTRL has a regret guarantee for the losses $\ell(\theta; d_t)$'s of form: (regret bound without the r_t's) $+\frac{1}{n}\sum_{t=1}^{n} r_t(\theta^*)$.

In the stochastic OCO setting (Section 4.2), we show that for least-square losses (where $\ell(\theta; d_t) = (y_t - \langle \mathbf{x}_t, \theta\rangle)^2$ with $d_t = (\mathbf{x}_t, y_t)$) and linear losses (when $\ell(\theta; d_t) = \langle d_t, \theta\rangle$), a variant of DP-FTRL achieves regret of the form $O\left(\left(\frac{1}{\sqrt{n}}+\frac{\sqrt{p}}{\varepsilon n}\right)\cdot \mathsf{polylog}\left(\frac{1}{\delta}, n, \frac{1}{\beta}\right)\right)$ with probability $1-\beta$ over the randomness of algorithm. Our guarantees are strictly high-probability guarantees, i.e., the regret only depends on $\mathsf{polylog}(1/\beta)$.

[4]Although not stated formally in the literature, a simple argument shows that locally private SGD (Duchi et al., 2013) can achieve the same regret as in (Smith & Thakurta, 2013).

Population risk guarantees: In Section 4.3, using the standard online-to-batch conversion (Cesa-Bianchi et al., 2002; Shalev-Shwartz et al., 2009), we obtain a population risk guarantee for DP-FTRL. For general Lipschitz convex losses, the population risk for DP-FTRL in Theorem C.5 is same as that in (Bassily et al., 2014, Appendix F) (up to logarithmic factors), but the advantage of DP-FTRL is that it is a single pass algorithm (over the data set D), as opposed to requiring n passes over the data. *Thus, we provide the best known population risk guarantee for a single pass algorithm that does not rely on convexity for privacy.* While the results in (Bassily et al., 2019a; 2020; Feldman et al., 2020a) have a tighter (and optimal) excess population risk of $\widetilde{\Theta}(1/\sqrt{n} + \sqrt{p}/(\varepsilon n))$, they either require convexity to ensure privacy for a single pass algorithm, or need to make n-passes over the data. For restricted classes like linear and least-squared losses, DP-FTRL can achieve the optimal population risk via the tighter stochastic regret guarantee. Whether DP-FTRL can achieve the optimal excess population risk in the general convex setting is left as an open problem.

Empirical contributions: In Section 5, we study some trade-offs between privacy/utility/computation for DP-FTRL and DP-SGD. We conduct our experiments on four benchmark data sets: MNIST, CIFAR-10, EMNIST, and StackOverflow. We start by fixing the computation available to the techniques, and observing privacy/utility trade-offs. We find that DP-FTRL achieves better utility compared to DP-SGD for moderate to large ε. In scenarios where amplification cannot be ensured (e.g., due to practical/implementation constraints), DP-FTRL provides substantially better performance as compared to unamplified DP-SGD. Moreover, we show that with a modest increase in the computation cost, DP-FTRL, without any need for amplification, can match the performance of amplified DP-SGD. Next, we focus on privacy/computation trade-offs for both the techniques when a utility target is desired. We show that DP-FTRL can provide better trade-offs compared to DP-SGD for various accuracy targets, which can result in

significant savings in privacy/computation cost as the size of data sets becomes limited.

To shed light on the empirical efficacy of DP-FTRL (in comparison) to DP-SGD, in Section 3.2, we show that a variant of DP-SGD (with correlated noise) can be viewed as an equivalent formulation of DP-FTRL in the unconstrained setting ($\mathcal{C} = \mathbb{R}^p$). In the case of traditional DP-SGD (Bassily et al., 2014), the scale of the noise added per-step $t \in [n]$ is asymptotically same as that of DP-FTRL once $t = \omega(n)$.

2. Background

Differential Privacy: Throughout the paper, we use the notion of approximate differential privacy (Dwork et al., 2006b;a) and Rényi differential privacy (RDP) (Abadi et al., 2016; Mironov, 2017). For meaningful privacy guarantees, ε is assumed to be a small constant, and $\delta \ll 1/|D|$.

Definition 2.1 (RDP (Abadi et al., 2016; Mironov, 2017))**.** *A randomized algorithm $\mathcal{A}$ is (α, ε)-RDP if for any pair of neighboring datasets D, D' that differ in one record, we have*

$$\frac{1}{\alpha - 1} \log \mathop{\mathbf{E}}_{o \sim \mathcal{A}(D)} \left(\frac{\mathbf{Pr}(\mathcal{A}(D) = o)}{\mathbf{Pr}(\mathcal{A}(D') = o)} \right)^{\alpha} \leq \varepsilon$$

Abadi et al. (2016) and Mironov (2017) have shown that an (α, ε)-RDP algorithm guarantees $\left(\varepsilon + \frac{\log(1/\delta)}{\alpha - 1}, \delta\right)$-differential privacy. Follow-up works (Asoodeh et al., 2020; Canonne et al., 2020) provide tighter conversions. We used the conversion in (Canonne et al., 2020) in our experiments.

To answer a query $f(D)$ with ℓ_2 sensitivity L, i.e., $\max_{\text{neighboring } D, D'} \|f(D) - f(D')\|_2 \leq L$, the Gaussian mechanism (Dwork et al., 2006b) returns $f(D) + \mathcal{N}(0, L^2\sigma^2)$, which guarantees $\left(\sqrt{1.25 \log(2/\delta)}/\sigma, \delta\right)$-differential privacy (Dwork et al., 2006b; Dwork & Roth, 2014) and $(\alpha, \alpha/2\sigma^2)$-RDP (Mironov, 2017).

DP-SGD and Privacy Amplification: Differentially-private stochastic gradient descent (DP-SGD) is a common algorithm to solve private optimization problems. The basic idea is to enforce a bounded ℓ_2 norm of individual gradient, and add Gaussian noise to the gradients used in SGD updates. Specifically, consider a dataset $D = \{d_1, \ldots, d_n\}$ and an objective function of the form $\sum_{i=1}^{n} \ell(\theta; d_i)$ for some loss function ℓ. DP-SGD uses an update rule

$$\theta_{t+1} \leftarrow \theta_t - \frac{\eta}{|\mathcal{B}|} \left(\sum_{i \in \mathcal{B}} \mathsf{clip}\left(\nabla_\theta \ell(\theta_t; d_i), L\right) + \mathcal{N}(0, L^2\sigma^2) \right)$$

where $\mathsf{clip}\,(v, L)$ projects v to the ℓ_2-ball of radius L, and $\mathcal{B} \subseteq [n]$ represents a mini-batch of data.

Using the analysis of the Gaussian mechanism, we know that such an update step guarantees $(\alpha, \alpha/2\sigma^2)$-RDP with respect to the mini-batch $\mathcal{B}$. By parallel composition, running one epoch with disjoint mini-batches guarantees $(\alpha, \alpha/2\sigma^2)$-RDP. On the other hand, previous works (Bassily et al., 2014; Abadi et al., 2016; Wang et al., 2019) has shown that if $\mathcal{B}$ is chosen uniformally at random from $[n]$, or if we use poisson sampling to collect a batch of samples $\mathcal{B}$, then one step would guarantee $\left(\alpha, O\left(\alpha/2\sigma^2 \cdot (|\mathcal{B}|/n)^2\right)\right)$-RDP.

Tree-based Aggregation: Consider the problem of privately releasing prefix sum of a data stream, i.e., given a stream $D = (d_1, d_2, \ldots, d_T)$ such that each $d_i \in \mathbb{R}^p$ has ℓ_2 norm bounded by L, we aim to release $s_t = \sum_{i=1}^{t} d_i$ for all $t \in [1, T]$ under differential privacy. Dwork et al. (2010); Chan et al. (2011) propose a tree-based aggregation algorithm to solve this problem. Consider a complete binary tree $\mathcal{T}$ with leaf nodes as d_1 to d_T, and internal nodes as the sum of all leaf nodes in its subtree. To release the exact prefix sum s_t, we only need to sum up $O\left(\log(t)\right)$ nodes. To guarantee differential privacy for releasing the tree $\mathcal{T}$, since any d_i appears in $\log(T)$ nodes in $\mathcal{T}$, using composition, we can add Gaussian noise of standard deviation of the order $L\sqrt{\log(T)\log(1/\delta)}/\varepsilon$ to guarantee (ε, δ)-differential privacy.

Smith & Thakurta (2013) used this aggregation algorithm to build a nearly optimal algorithms for private online learning. One important aspect of Smith & Thakurta (2013) result is that it showed the privacy guarantee holds even for *adaptively chosen sequences* $\{d_t\}_{t=1}^{T}$, which is crucial for model training tasks.

3. Private Follow-The-Regularized-Leader

In this section, we provide the formal description of the DP-FTRL algorithm (Algorithm 1) and its privacy analysis. We then show that a variant of differentially private stochastic gradient descent (DP-SGD) (Song et al., 2013; Bassily et al., 2014) can be viewed of as an instantiation of DP-FTRL under appropriate choice of learning rate.

Critically, *our privacy guarantees for DP-FTRL hold when the data D are processed in an arbitrary (even adversarily chosen) order,* and do not depend on the convexity of the loss functions. The utility guarantees, i.e., the regret and the excess risk guarantees require convex losses (i.e., $\ell(\cdot;\cdot)$ is convex in the first parameter). In the presentation below, we assume differentiable losses for brevity. The arguments extend to non-differentiable convex losses via standard use of sub-differentials (Shalev-Shwartz et al., 2011; Hazan, 2019).

3.1. Algorithm Description

The main idea of DP-FTRL is based on three observations: i) For online convex optimization, to bound the regret, for a given loss function $\ell(\theta; d_t)$ (i.e., the loss at time step t), it suffices for the algorithm to operate on a linearization of the loss at θ_t (the model output at time step t): $\widetilde{\ell}(\theta; d_t) = \langle \nabla_\theta \ell(\theta_t; d_t), \theta - \theta_t \rangle$, ii) Under appropriate choice of λ, optimizing for $\theta_{t+1} = \arg\min_{\theta \in \mathcal{C}} \sum_{i=1}^{t} \widetilde{\ell}(\theta; d_t) + \frac{\lambda}{2} \|\theta\|_2^2$ over $\theta \in \mathcal{C}$ gives a good model at step $t+1$, and iii) For all $t \in [n]$, one can privately keep track of $\sum_{i=1}^{t} \widetilde{\ell}(\theta; d_t)$ using the now standard *tree aggregation protocol* (Dwork et al., 2010; Chan et al., 2011). While a variant of this idea was used in (Smith & Thakurta, 2013) under the name of *follow-the-approximate-leader*, one key difference is that they used a quadratic approximation of the regularized loss, i.e., $\ell(\theta; d_t) + \frac{\lambda}{t} \|\theta\|_2^2$. This formulation results in a more complicated algorithm, sub-optimal regret analysis, and failure to maintain structural properties (like sparsity) introduced by composite losses (Duchi et al., 2010; Xiao, 2010; McMahan, 2011; 2017).

Algorithm 1 $\mathcal{A}_{\mathsf{FTRL}}$: Differentially Private Follow-The-Regularized-Leader (DP-FTRL)

Require: Data set: $D = [d_1, \cdots, d_n]$ arriving in a stream, in an arbitrary order; constraint set: $\mathcal{C}$, noise scale: σ, regularization parameter: λ, clipping norm: L.

1: $\theta_1 \leftarrow \arg\min_{\theta \in \mathcal{C}} \frac{\lambda}{2} \|\theta\|_2^2$. **Output** θ_1.
2: $\mathcal{T} \leftarrow \texttt{InitializeTree}\,(n, \sigma^2, L)$.
3: **for** $t \in [n]$ **do**
4: Let $\nabla_t \leftarrow \mathsf{clip}\,(\nabla_\theta \ell(\theta_t; d_t), L)$, where $\mathsf{clip}\,(\boldsymbol{v}, L) = \boldsymbol{v} \cdot \min\left\{\frac{L}{\|\boldsymbol{v}\|_2}, 1\right\}$.
5: $\mathcal{T} \leftarrow \texttt{AddToTree}\,(\mathcal{T}, t, \nabla_t)$.
6: $\boldsymbol{s}_t \leftarrow \texttt{GetSum}\,(\mathcal{T}, t)$, i.e., estimate $\sum_{i=1}^{t} \nabla_i$ via tree-aggregation protocol.
7: $\theta_{t+1} \leftarrow \arg\min_{\theta \in \mathcal{C}} \langle \boldsymbol{s}_t, \theta \rangle + \frac{\lambda}{2} \|\theta\|_2^2$. **Output** θ_{t+1}.
8: **end for**

Later in the paper, we provide two variants of DP-FTRL (momentum DP-FTRL, and DP-FTRL for least square losses) which will have superior privacy/utility trade-offs for certain problem settings.

DP-FTRL is formally described in Algorithm 1. There are three functions, `InitializeTree`, `AddToTree`, `GetSum`, that correspond to the tree-aggregation algorithm. At a high-level, `InitializeTree` initializes the tree data structure $\mathcal{T}$, `AddToTree` allows adding a new gradient ∇_t to $\mathcal{T}$, and `GetSum` returns the prefix sum $\sum_{i=1}^{t} \nabla_t$ privately. In our experiments (Section 5), we use the iterative estimator from (Honaker, 2015) to obtain the optimal estimate of the prefix sums in `GetSum`. Please refer to Appendix B.1 for the formal algorithm descriptions.

It can be shown that the error introduced in DP-FTRL due to privacy is dominated by the error in estimating $\sum_{i=1}^{t} \nabla_t$ at each $t \in [n]$. It follows from (Smith & Thakurta, 2013) that for a sequence of (adaptively chosen) vectors $\{\nabla_t\}_{t=1}^{n}$, if we perform $\texttt{AddToTree}\,(\mathcal{T}, t, \nabla_t)$ for each $t \in [n]$, then we can write $\texttt{GetSum}\,(\mathcal{T}, t) = \sum_{i=1}^{t} \nabla_i + \boldsymbol{b}_t$ where $\boldsymbol{b}_t$ is normally distributed with mean zero, and $\forall t \in [n], \|\boldsymbol{b}_t\|_2 \leq L\sigma\sqrt{p\lceil \lg(n) \rceil \ln(n/\beta)}$ w.p. at least $1 - \beta$.

Momentum Variant: We find that using a momentum term $\gamma \in [0, 1]$ with Line 7 in Algorithm 1 replaced by

$$\boldsymbol{v}_t \leftarrow \gamma \cdot \boldsymbol{v}_{t-1} + \boldsymbol{s}_t,\ \theta_{t+1} \leftarrow \arg\min_{\theta \in \mathcal{C}} \langle \boldsymbol{v}_t, \theta \rangle + \frac{\lambda}{2} \|\theta - \theta_0\|_2^2$$

gives superior empirical privacy/utility trade-off compared to the original algorithm when training non-convex models. Throughout the paper, we refer to this variant as momentum DP-FTRL, or DP-FTRLM. Although we do not provide formal regret guarantee for this variant, we conjecture that the superior empirical performance is due to the following reason. The noise added by the tree aggregation algorithm is always bounded by $O(\sqrt{p \ln(1/\delta)} \cdot \ln(n)/\varepsilon)$. However, the noise at time step t and $t+1$ can differ by a factor of $O(\sqrt{\ln n})$. This creates sudden jumps in between the output models comparing to DP-SGD. The momentum can smooth out these jumps.

Privacy analysis: In Theorem 3.1, we provide the privacy guarantee for Algorithm 1 and its momentum variant (with proof in Appendix B.2). In Appendix D, we extend it to multiple passes over the data set D, and batch sizes > 1.

Theorem 3.1 (Privacy guarantee)**.** *If $\|\nabla_\theta \ell(\theta; d)\|_2 \leq L$ for all $d \in \mathcal{D}$ and $\theta \in \mathcal{C}$, then Algorithm 1 (and its momentum variant) guarantees $\left(\alpha, \frac{\alpha\lceil \lg(n) \rceil}{2\sigma^2}\right)$-Rényi differential privacy, where n is the number of samples in D. Setting $\sigma = \frac{\sqrt{2\lceil \lg(n) \rceil \ln(1/\delta)}}{\varepsilon}$, one can guarantee (ε, δ)-differential privacy, for $\varepsilon \leq 2\ln(1/\delta)$.*

DP-FTRL's memory footprint as compared to DP-SGD: At any given iteration, the cost of computing the mini-batch gradients is exactly the same for both DP-FTRL and DP-SGD. The only difference between the memory usage of DP-FTRL as compared to DP-SGD is that DP-FTRL needs to keep track of worst-case $(\log_2(t) + 2)$ past gradient information for iteration t. Note that these are precomputed objects that can be stored in memory.

3.2. Comparing Noise in DP-SGD and DP-FTRL

In this section, we use the equivalence of non-private SGD and FTRL (McMahan, 2017) to establish equivalence between a variant of noisy-SGD and DP-FTRL, and hence make DP-SGD and DP-FTRl comparable.

Let $D = \{d_1, \ldots, d_n\}$ be the data set of size n. Consider a general noisy-SGD algorithm with update rule $\theta_{t+1} \leftarrow \theta_t - \eta \cdot (\nabla_\theta \ell(\theta_t; d_t) + \boldsymbol{a}_t)$, where η is the learning rate and $\boldsymbol{a}_t$ is some random noise. DP-SGD can be viewed as a special case, where d_t is sampled u.a.r. from D, and $\boldsymbol{a}_t$ is drawn i.i.d. from $\mathcal{N}\left(0, \widetilde{O}\left(\frac{L^2}{n\varepsilon^2}\right)\right)$. If we expand the recursive relation, we can see that the total amount of noise added to the estimation of θ_{t+1} is $\eta \sum_{i=1}^{t} \boldsymbol{a}_t = \mathcal{N}\left(0, \widetilde{O}\left(\frac{\eta^2 L^2 t}{n\varepsilon^2}\right)\right)$. Define $\boldsymbol{b}_0 = 0$, and let $\boldsymbol{b}_t$ be the noise added by the tree-aggregation algorithm at time step t of Algorithm $\mathcal{A}_{\mathsf{FTRL}}$. We can show that DP-FTRL can be written in the same form as in the above general noisy-SGD formula, where i) the noise $\boldsymbol{a}_t = \boldsymbol{b}_t - \boldsymbol{b}_{t-1}$, ii) the data samples d_t's are drawn in sequence from D, and iii) the learning rate η is set to be $\frac{1}{\lambda}$, where λ is the regularization parameter in Algorithm $\mathcal{A}_{\mathsf{FTRL}}$. In this variant of noisy SGD, the total noise added to the model is $\boldsymbol{b}_t = \mathcal{N}\left(0, \widetilde{O}\left(\frac{\eta^2 \cdot L^2}{\varepsilon^2}\right)\right)$.

Under the same form of the update rule, we can roughly (as the noise is not independent in the DP-FTRL case) compare the two algorithms. When $t = \Omega(n)$, the noise of DP-SGD *with amplification* matches that of DP-FTRL up to factor of $\mathsf{polylog}(n)$. As a result, we expect (and as corroborated by the population risk guarantees and experiments) sampled DP-SGD and DP-FTRL to perform similarly. (In Appendix B.3 we provide a formal equivalence.)

4. Regret and Population Risk Guarantees

In this section we consider the setting when loss function ℓ is convex in its first parameter, and provide for DP-FTRL: i) Adversarial regret guarantees for general convex losses, ii) Tighter stochastic regret guarantees for least-squares and linear losses, and iii) Population risk guarantees via online-to-batch conversion. All our guarantees are high-probability over the randomness of the algorithm, i.e., w.p. at least $1 - \beta$, the error only depends on $\mathsf{polylog}(1/\beta)$.

4.1. Adversarial Regret for (Composite) Losses

The theorem here gives a regret guarantee for Algorithm 1 against a *fully adaptive* (Shalev-Shwartz et al., 2011) adversary who chooses the loss function $\ell(\theta; d_t)$ based on $[\theta_1, \ldots, \theta_t]$, but without knowing the internal randomness of the algorithm. See Appendix C.1 for a more general version of Theorem 4.1, and its proof.

Theorem 4.1 (Regret guarantee)**.** *Let θ be any model in $\mathcal{C}$, $[\theta_1, \ldots, \theta_n]$ be the outputs of Algorithm $\mathcal{A}_{\mathsf{FTRL}}$ (Algorithm 1), and let L be a bound on the ℓ_2-Lipschitz constant of the loss functions. Setting λ optimally and plugging in the noise scale σ from Theorem 3.1 to ensure (ε, δ)-differential privacy, we have that for any $\theta^* \in \mathcal{C}$, w.p. at least $1 - \beta$ over the randomness of $\mathcal{A}_{\mathsf{FTRL}}$, the regret*

$$R_D(\mathcal{A}_{\mathsf{FTRL}}; \theta^*) = O\left(L \left\|\theta^*\right\|_2 \cdot \left(\frac{1}{\sqrt{n}} + \sqrt{\frac{p^{1/2} \ln^2(1/\delta) \ln(1/\beta)}{\varepsilon n}}\right)\right).$$

Extension to composite losses: Composite losses (Duchi et al., 2010; McMahan, 2011; 2017) refer to the setting where in each round, the algorithm is provided with a function $f_t(\theta) = \ell(\theta; d_t) + r_t(\theta)$ with $r_t : \mathcal{C} \to \mathbb{R}^+$ being a convex regularizer that does not depend on the data sample d_t. The ℓ_1-regularizer, $r_t(\theta) = \|\theta\|_1$, is perhaps the most important practical example, playing a critical role in high-dimensional statistics (e.g., in the LASSO method) (Bhlmann & van de Geer, 2011), as well as for applications like click-through-rate (CTR) prediction where very sparse models are needed for efficiency (McMahan et al., 2013). In order to operate on composite losses, we simply replace Line 7 of Algorithm $\mathcal{A}_{\mathsf{FTRL}}$ with

$$\theta_{t+1} \leftarrow \arg\min_{\theta \in \mathcal{C}} \langle \boldsymbol{s}_t, \theta \rangle + \sum_{i=1}^{t} r_i(\theta) + \frac{\lambda}{2} \|\theta\|_2^2,$$

which can be solved in closed form in many important cases such as ℓ_1 regularization. We obtain Corollary 4.2, analogous to (McMahan, 2017, Theorem 1) in the non-private case. We do not require any assumption (e.g., Lipschitzness) on the regularizers beyond convexity since we *only linearize the losses* in Algorithm $\mathcal{A}_{\mathsf{FTRL}}$. It is worth mentioning that (Smith & Thakurta, 2013) is fundamentally incompatible with this type of guarantee.

Corollary 4.2. *Let θ be any model in $\mathcal{C}$, $[\theta_1, \ldots, \theta_n]$ be the outputs of Algorithm $\mathcal{A}_{\mathsf{FTRL}}$ (Algorithm 1), and L be a bound on the ℓ_2-Lipschitz constant of the loss functions. W.p. at least $1-\beta$ over the randomness of the algorithm, for any $\theta^* \in \mathcal{C}$, assuming $\mathbf{0} \in \mathcal{C}$, we have: $R_D(\mathcal{A}_{\mathsf{FTRL}}; \theta^*) \leq$*

$$\frac{L\sigma\sqrt{p\lceil \lg n \rceil \ln(n/\beta)} + L^2}{\lambda} + \frac{\lambda}{2n} \|\theta^*\|_2^2 + \frac{1}{n} \sum_{t=1}^{n} r_t(\theta^*).$$

4.2. Stochastic Regret for Least-squared Losses

In this setting, for each data sample $d_i = (\mathbf{x}_i, y_i)$ (with $\mathbf{x}_i \in \mathbb{R}^p$ and $y_i \in \mathbb{R}$) in the data set $D = \{d_1, \ldots, d_n\}$, the corresponding loss takes the least-squares form[5]:

[5] A similar argument as in Theorem 4.3 can be used in the setting where the loss functions are linear, $\ell(\theta; d) = \langle \theta, d \rangle$ with $d \in \mathbb{R}^p$ and $\|d\|_2 \leq L$.

$\ell(\theta; d_i) = (y_i - \langle \mathbf{x}_i, \theta \rangle)^2$. We also assume that each data sample d_i is drawn i.i.d. from some fixed distribution τ.

A straightforward modification of DP-FTRL, $\mathcal{A}_{\text{FTRL-LS}}$ (Algorithm 2 in Appendix C.2), achieves the following.

Theorem 4.3 (Stochastic regret for least-squared losses)**.** *Let $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\} \in \mathcal{D}^n$ be a data set drawn i.i.d. from τ, let $L = \max_{\mathbf{x} \in \mathcal{D}} \|\mathbf{x}\|_2$, and let $\max_{y \sim \mathcal{D}} |y| \leq 1$. Let $\theta^* \in \mathcal{C}$, $\mu = \max_{\theta \in \mathcal{C}} \|\theta\|_2$, and $\rho = \max\{\mu, \mu^2\}$. Then $\mathcal{A}_{\text{FTRL-LS}}$ provides (ε, δ)-differentially privacy while outputting $[\theta_1, \ldots, \theta_n]$ s.t. w.p. at least $1 - \beta$ for any $\theta^* \in \mathcal{C}$, $\mathbb{E}_D [R_D(\mathcal{A}_{\text{FTRL-LS}}; \theta^*)] =$*

$$O\left(L^2 \rho^2 \left(\sqrt{\frac{\ln(n)}{n}} + \frac{\sqrt{p \ln^5(n/\beta) \cdot \ln(1/\delta)}}{\varepsilon n}\right)\right).$$

The arguments of (Agarwal & Singh, 2017) can be extended to show a similar regret guarantee *in expectation only*, whereas ours is a high-probability guarantee.

4.3. Excess Risk via Online-to-Batch Conversion

Using the online-to-batch conversion (Cesa-Bianchi et al., 2002; Shalev-Shwartz et al., 2009), from Theorem 4.1, we can obtain a population risk guarantee $O\left(\left(\sqrt{\frac{\ln(1/\beta)}{n}} + \sqrt{\frac{p^{1/2} \ln^2(1/\delta) \ln(1/\beta)}{\varepsilon n}}\right)\right)$, where β is the failure probability. (See Appendix C.3 for a formal statement.) For least squares and linear losses, using the regret guarantee in Theorem 4.3 and online-to-batch conversion, one can actually achieve the optimal population risk (up to logarithic factors) $O\left(\sqrt{\frac{\ln(n) \ln(1/\beta)}{n}} + \frac{\sqrt{p \ln^5(n/\beta) \cdot \ln(1/\delta)}}{\varepsilon n}\right)$.

5. Empirical Evaluation

We provide an empirical evaluation of DP-FTRL on four benchmark data sets, and compare its performance with the state-of-the-art DP-SGD on three axes: (1) **Privacy**, measured as an (ε, δ)-DP guarantee on the mechanism, (2) **Utility**, measured as (expected) test set accuracy for the trained model under the DP guarantee, and (3) **Computation cost**, which we measure in terms of mini-batch size and number of training iterations. The code is open sourced[6].

First, we evaluate the privacy/utility trade-offs provided by each technique at fixed computation costs. Second, we evaluate the privacy/computation trade-offs each technique can provide at fixed utility targets. A natural application for this is distributed frameworks such as FL, where the privacy budget and a desired utility threshold can be fixed, and the goal is to satisfy both constraints with the least computation. Computational cost is of critical importance in FL, as it can get challenging to find available clients with increasing mini-batch size and/or number of training rounds.

We show the following results: (1) DP-FTRL provides superior privacy/utility trade-offs than unamplified DP-SGD, (2) For a modest increase in computation cost, DP-FTRL (that does not use any privacy amplification) can match the privacy/utility trade-offs of amplified DP-SGD for all privacy regimes, and further (3) For regimes with large privacy budgets, DP-FTRL achieves higher accuracy than amplified DP-SGD even at the same computation cost, (4) For realistic data set sizes, DP-FTRL can provide superior privacy/computation trade-offs compared to DP-SGD.

5.1. Experimental Setup

Datasets: We conduct our evaluation on three image classification tasks, MNIST (LeCun et al., 1998), CIFAR-10 (Krizhevsky, 2009), EMNIST (ByMerge split) (Cohen et al., 2017); and a next word prediction task on StackOverflow data set (Overflow, 2018). Since StackOverflow is naturally keyed by users, we assume training in a federated learning setting, i.e., using the Federated Averaging optimizer for training over users in StackOverflow. The privacy guarantee is thus user-level, in contrast to the example-level privacy for the other three datasets (see Definition 1.1).

For all experiments with DP, we set the privacy parameter δ to 10^{-5} on MNIST and CIFAR-10, and 10^{-6} on EMNIST and StackOverflow, s.t. $\delta < n^{-1}$, where n is the number of users in StackOverflow (or examples in the other data sets).

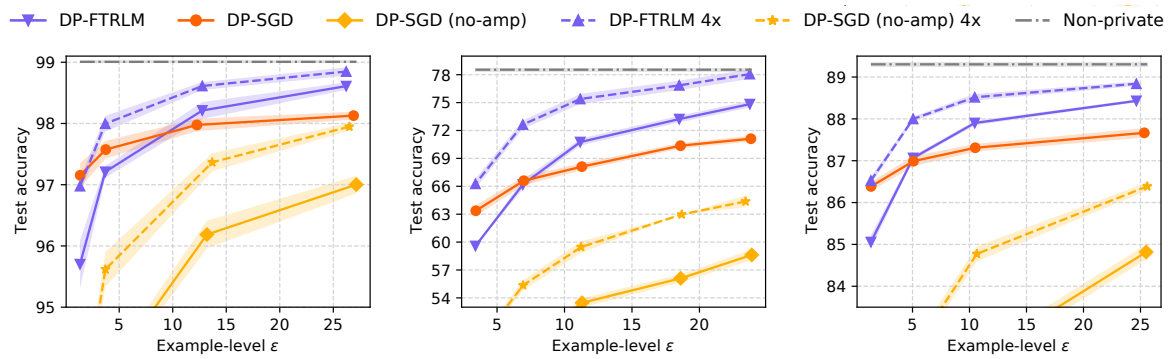
Model Architectures: For all the image classification tasks, we use small convolutional neural networks as in prior work (Papernot et al., 2020b). For StackOverflow, we use the one-layer LSTM network described in (Reddi et al., 2020). See Appendix E.1 for more details.

Optimizers: We consider DP-FTRL with mini-batch model updates, and multiple epochs. We provide a privacy analysis for both the extensions in Appendix D. We also consider its momentum variant DP-FTRLM. We find that DP-FTRLM with momentum 0.9 always outperforms DP-FTRL. Similarly, for DP-SGD (Google, 2019), we consider its momentum variant (DP-SGDM), and report the best-performing variant in each task. See Appendix E.2 for a comparison of the two optimizers for both techniques.

5.2. Privacy/Utility Trade-offs with Fixed Computation

In Figure 1, we show accuracy / privacy tradeoffs (by varying the noise multiplier) at fixed computation costs. Since

[6] https://github.com/google-research/federated/tree/master/dp_ftrl for FL experiments, and https://github.com/google-research/DP-FTRL for centralized learning.



(a) MNIST (b) CIFAR-10 (c) EMNIST (ByMerge)

Figure 1. Privacy/accuracy trade-offs for DP-SGD (private baseline), DP-SGD without amplification (label "DP-SGD (no-amp)"), and DP-FTRLM on MNIST (mini-batch size 250), CIFAR-10 (mini-batch size 500), and EMNIST (mini-batch size 500). "4x" in the label denotes four times computation cost (by increasing batch size four times). Results for "DP-SGD 4x" are deferred to Appendix F.

both DP-FTRL and DP-SGD require clipping gradients from each sample and adding noise to the aggregated update in each iteration, we consider the number of iterations and the minibatch size as a proxy for computation cost. For each experiment, we run five independent trials, and plot the mean and standard deviation of the final test accuracy at different privacy levels. We provide details of hyperparameter tuning for all the techniques in Appendix F.1.

DP-SGD is the state-of-the-art technique used for private deep learning, and amplification by subsampling (or shuffling) forms a crucial component in its privacy analysis. Thus, we take amplified DP-SGD (or its momentum variant when performance is better) at a fixed computation cost as our baseline. We fix the (samples in mini-batch, training iterations) to (250, 4800) for MNIST, (500, 10000) for CIFAR-10, and (500, 69750) for EMNIST. Our goal is to achieve equal or better tradeoffs while processing data in an arbitrary order (i.e., without relying on any amplification).

DP-SGD without any privacy amplification ("DP-SGD (no-amp)") cannot achieve this: For all the data sets, the accuracy with DP-SGD (no-amp) at the highest ε in Figure 1 is worse than the accuracy of the DP-SGD baseline even at its lowest ε. Further, if we increase the computation by four times (increasing the mini-batch size by four times), the privacy/utility trade-offs of "DP-SGD (no-amp) 4x" are still substantially worse than the private baseline.[7]

For DP-FTRLM at the same computation cost as our DP-SGD baseline, as the privacy parameter ε increases, the relative performance of DP-FTRLM improves for each data set, even outperforming the baseline for larger values of ε. Further, if we increase the batch size by four times for DP-FTRLM, its privacy-utility trade-off almost always matches or outperforms the amplified DP-SGD baseline, affirmatively answering this paper's primary question. In particular, for CIFAR-10 (Figure 1b), "DP-FTRLM 4x" provides superior performance than the DP-SGD baseline even for the lowest ε.

We observe similar results for StackOverflow with user-level DP in Figure 2a. We fix the computation cost to 100 clients per round (also referred to as the report goal), and 1600 training rounds. DP-SGDM (or more precisely in this case, DP-FedAvg with server momentum) is our baseline. For DP-SGDM without privacy amplification (DP-SGDM no-amp), the privacy/accuracy trade-off never matches that of the DP-SGDM baseline, and gets significantly worse for lower ε. With a 4x increase in report goal, DP-SGDM no-amp nearly matches the privacy/utility trade-off of the DP-SGD baseline, outperforming it for larger ε.

For DP-FTRLM, with the same computation cost as the DP-SGDM baseline, it outperforms the baseline for the larger ε, whereas for the four-times increased report goal, it provides a strictly better privacy/utility trade-off. We conclude DP-FTRL provides superior privacy/utility trade-offs than unamplified DP-SGD, and for a modest increase in computation cost, it can match the performance of DP-SGD, without the need for privacy amplification.

5.3. Privacy/Computation Trade-offs with Fixed Utility

For a sufficiently large data set / population, better privacy vs. accuracy trade-offs can essentially always be achieved at the cost of increased computation. Thus, in this section we slice the privacy/utility/computation space by fixing utility (accuracy) targets, and evaluating how much computation (report goal) is necessary to achieve different ε for StackOverflow. Our non-private baseline achieves

[7] For completeness, we provide plots with the full performance of DP-SGD (no-amp), DP-SGD (no-amp) 4x, and DP-SGD 4x, in Appendix F.2.

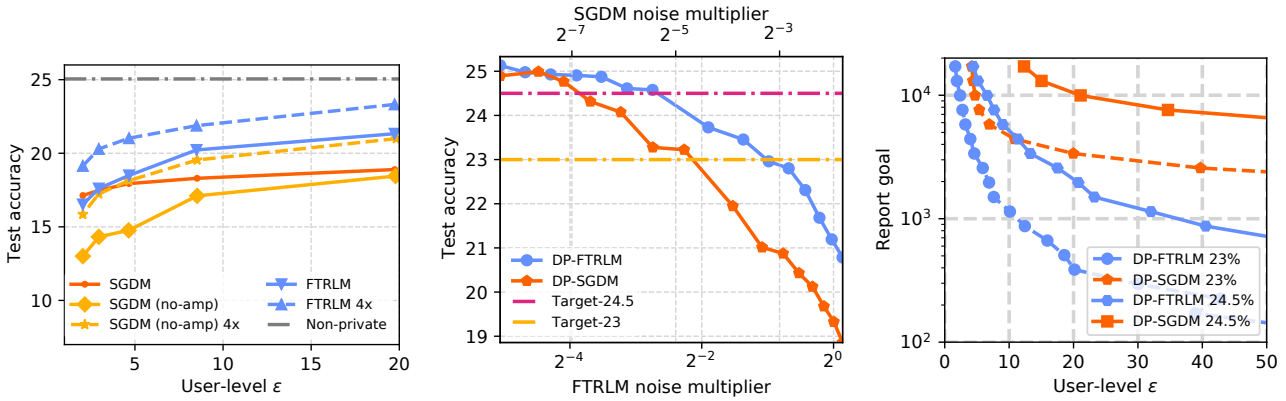


Figure 2. (a) Accuracy on StackOverflow under different privacy epsilon by varying noise multiplier and batch sizes. (b) Test accuracy of DP-SGDM and DP-FTRLM with various noise multipliers for StackOverflow. (c)Relationship between user-level privacy ε (when $\delta \approx 1/\text{population}$) and computation cost (report goal) for two fixed accuracy targets (see legend) on the StackOverflow data set.

an accuracy of 25.15%, and we fix 24.5% (2.6% relative loss) and 23% (8.6% relative loss) as our accuracy targets. Note that from the accuracy-privacy trade-offs presented in Figure 2a, achieving even 23% for either DP-SGD or DP-FTRL will result in a large ε for the considered report goals.

For each target, we tune hyperparameters (see Appendix G.2 for details) for both DP-SGDM and DP-FTRLM at a fixed computation cost to obtain the maximum noise scale for each technique while ensuring the trained models meet the accuracy target. Specifically, we fix a report goal of 100 clients per round for 1600 training rounds, and tune DP-SGD and DP-FTRL for 15 noise multipliers, ranging from $(0, 0.3)$ for DP-SGD, and $(0, 1.13)$ for DP-FTRL. At this report goal, for noise multiplier 0.3, DP-SGD provides 18.89% accuracy at $\varepsilon \sim 19$, whereas for noise multiplier 1.13 DP-FTRL provides 19.74% accuracy at $\varepsilon \sim 19$. We provide the results in Figure 2b.

For each target accuracy, we choose the largest noise multiplier for each technique that results in the trained model achieving the accuracy target. For accuracies (23%, 24.5%), we select noise multipliers (0.015, 0.007) for DP-SGDM, and (0.387, 0.149) for DP-FTRLM, respectively. This data allows us to evaluate the privacy/computation trade-offs for both techniques, assuming the accuracy stays constant as we scale up the noise and report goal together (maintaining a constant signal-to-noise ratio while improving ε). This assumption was introduced and validated by (McMahan et al., 2017b), which showed that keeping the clipping norm bound, training rounds, and the scale of the noise added to the model update constant, increasing the report goal does not change the final model accuracy. In Appendix G.1, we independently corroborate this effect for both DP-SGD and DP-FTRL on StackOverflow.

We plot the results in Figure 2c. For both the accuracy targets, DP-FTRLM achieves any privacy $\varepsilon \in (0, 50)$ at a lower computational cost than DP-SGDM. In Appendix G.3, we provide a similar plot for a hypothetically larger population, where we see that DP-FTRLM provides superior performance than DP-SGDM for most of the considered privacy regimes.

6. Conclusion

In this paper we introduce the DP-FTRL algorithm, which we show to have the tightest known regret guarantees under DP, and have the best known excess population risk guarantees for a single pass algorithm on non-smooth convex losses. For linear and least-squared losses, we show DP-FTRL actually achieves the optimal population risk. Furthermore, we show on benchmark data sets that DP-FTRL, which does not rely on any privacy amplification, can outperform amplified DP-SGD at large values of ε, and be competitive to it for all ranges of ε for a modest increase in computation cost (batch size). This work leaves two main open questions: i) Can DP-FTRL achieve the optimal excess population risk for all convex losses in a single pass?, and ii) Can one tighten the empirical gap between DP-SGD and DP-FTRL at smaller values of ε, possibly via a better estimator of the gradient sums from the tree data structure?

Acknowledgements

We would like to thank Borja Balle and Satyen Kale for the helpful discussions through the course of this project. We would also like to thank Adam Smith for suggesting the use of (Honaker, 2015) for variance reduction.

References

Abadi, M., Chu, A., Goodfellow, I. J., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *Proc. of the 2016 ACM SIGSAC Conf. on Computer and Communications Security (CCS'16)*, pp. 308–318, 2016.

Abernethy, J., Jung, Y. H., Lee, C., McMillan, A., and Tewari, A. Online learning via the differential privacy lens. In *NeurIPS*, 2019.

Agarwal, N. and Singh, K. The price of differential privacy for online learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 32–40, 2017.

Asoodeh, S., Liao, J., Calmon, F. P., Kosut, O., and Sankar, L. A better bound gives a hundred rounds: Enhanced privacy guarantees via f-divergences. In *2020 IEEE International Symposium on Information Theory (ISIT)*, pp. 920–925. IEEE, 2020.

Balle, B., Kairouz, P., McMahan, B., Thakkar, O. D., and Thakurta, A. Privacy amplification via random checkins. *Advances in Neural Information Processing Systems*, 33, 2020.

Bassily, R., Smith, A., and Thakurta, A. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *Proc. of the 2014 IEEE 55th Annual Symp. on Foundations of Computer Science (FOCS)*, pp. 464–473, 2014.

Bassily, R., Feldman, V., Talwar, K., and Thakurta, A. Private stochastic convex optimization with optimal rates. In *Advances in Neural Information Processing Systems*, pp. 11279–11288, 2019a.

Bassily, R., Feldman, V., Talwar, K., and Thakurta, A. G. Private stochastic convex optimization with optimal rates. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pp. 11279–11288, 2019b.

Bassily, R., Feldman, V., Guzmán, C., and Talwar, K. Stability of stochastic gradient descent on nonsmooth convex losses. *arXiv preprint arXiv:2006.06914*, 2020.

Bhlmann, P. and van de Geer, S. *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer Publishing Company, Incorporated, 2011. ISBN 3642201911.

Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konecny, J., Mazzocchi, S., McMahan, H. B., et al. Towards federated learning at scale: System design. *arXiv preprint arXiv:1902.01046*, 2019.

Canonne, C., Kamath, G., and Steinke, T. The discrete gaussian for differential privacy. *arXiv preprint arXiv:2004.00010*, 2020.

Cesa-Bianchi, N., Conconi, A., and Gentile, C. On the genomeralization ability of on-line learning algorithms. In *Advances in neural information processing systems*, pp. 359–366, 2002.

Chan, T.-H. H., Shi, E., and Song, D. Private and continual release of statistics. *ACM Trans. on Information Systems Security*, 14(3):26:1–26:24, November 2011.

Chaudhuri, K., Monteleoni, C., and Sarwate, A. D. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(Mar):1069–1109, 2011.

Cohen, G., Afshar, S., Tapson, J., and Schaik, A. V. Emnist: Extending mnist to handwritten letters. *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017. doi: 10.1109/ijcnn.2017.7966217.

Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

Duchi, J. C., Shalev-Shwartz, S., Singer, Y., and Tewari, A. Composite objective mirror descent. In *COLT*, pp. 14–26. Citeseer, 2010.

Duchi, J. C., Jordan, M. I., and Wainwright, M. J. Local privacy and statistical minimax rates. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pp. 429–438. IEEE Computer Society, 2013. doi: 10.1109/FOCS.2013.53. URL https://doi.org/10.1109/FOCS.2013.53.

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

Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., and Naor, M. Our data, ourselves: Privacy via distributed noise generation. In *Advances in Cryptology—EUROCRYPT*, pp. 486–503, 2006a.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In *Proc. of the Third Conf. on Theory of Cryptography (TCC)*, pp. 265–284, 2006b. URL http://dx.doi.org/10.1007/11681878_14.

Dwork, C., Naor, M., Pitassi, T., and Rothblum, G. N. Differential privacy under continual observation. In *Proc. of the Forty-Second ACM Symp. on Theory of Computing (STOC'10)*, pp. 715–724, 2010.

Erlingsson, Ú., Feldman, V., Mironov, I., Raghunathan, A., Talwar, K., and Thakurta, A. Amplification by shuffling: From local to central differential privacy via anonymity. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 2468–2479. SIAM, 2019.

Erlingsson, Ú., Feldman, V., Mironov, I., Raghunathan, A., Song, S., Talwar, K., and Thakurta, A. Encode, shuffle, analyze privacy revisited: Formalizations and empirical evaluation. *CoRR*, abs/2001.03618, 2020.

Evfimievski, A., Gehrke, J., and Srikant, R. Limiting privacy breaches in privacy preserving data mining. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pp. 211–222, 2003.

Facebook. Introducing opacus: A high-speed library for training pytorch models with differential privacy, 2020.

Feldman, V., Mironov, I., Talwar, K., and Thakurta, A. Privacy amplification by iteration. In *59th Annual IEEE Symp. on Foundations of Computer Science (FOCS)*, pp. 521–532, 2018.

Feldman, V., Koren, T., and Talwar, K. Private stochastic convex optimization: Optimal rates in linear time. In *Proc. of the Fifty-Second ACM Symp. on Theory of Computing (STOC'20)*, 2020a.

Feldman, V., McMillan, A., and Talwar, K. Hiding among the clones: A simple and nearly optimal analysis of privacy amplification by shuffling. *arXiv preprint arXiv:2012.12803*, 2020b.

Google. Tensorflow-privacy. https://github.com/tensorflow/privacy, 2019.

Hazan, E. Introduction to online convex optimization. *arXiv preprint arXiv:1909.05207*, 2019.

Hazan, E. and Kale, S. Beyond the regret minimization barrier: optimal algorithms for stochastic strongly-convex optimization. *The Journal of Machine Learning Research*, 15(1):2489–2512, 2014.

Honaker, J. Efficient Use of Differentially Private Binary Trees. In *Theory and Practice of Differential Privacy (TPDP 2015), London, UK*, 2015.

Iyengar, R., Near, J. P., Song, D., Thakkar, O., Thakurta, A., and Wang, L. Towards practical differentially private convex optimization. In *2019 IEEE Symposium on Security and Privacy (SP)*, 2019.

Jagielski, M., Ullman, J., and Oprea, A. Auditing differentially private machine learning: How private is private sgd? *arXiv preprint arXiv:2006.07709*, 2020.

Jain, P. and Thakurta, A. G. (near) dimension independent risk bounds for differentially private learning. In *International Conference on Machine Learning*, pp. 476–484, 2014.

Jain, P., Kothari, P., and Thakurta, A. Differentially private online learning. In *Proc. of the 25th Annual Conf. on Learning Theory (COLT)*, volume 23, pp. 24.1–24.34, June 2012.

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. *arXiv preprint arXiv:1912.04977*, 2019.

Kalai, A. and Vempala, S. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

Kasiviswanathan, S. P., Lee, H. K., Nissim, K., Raskhodnikova, S., and Smith, A. D. What can we learn privately? In *49th Annual IEEE Symp. on Foundations of Computer Science (FOCS)*, pp. 531–540, 2008.

Kifer, D., Smith, A., and Thakurta, A. Private convex empirical risk minimization and high-dimensional regression. In *Conference on Learning Theory*, pp. 25–1, 2012.

Krizhevsky, A. Learning multiple layers of features from tiny images, 2009.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

McMahan, B. Follow-the-regularized-leader and mirror descent: Equivalence theorems and l1 regularization. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 525–533, 2011.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, pp. 1273–1282, 2017a. URL http://proceedings.mlr.press/v54/mcmahan17a.html.

McMahan, H. B. A survey of algorithms and analysis for adaptive online learning. *Journal of Machine Learning Research*, 18(90):1–50, 2017. URL http://jmlr.org/papers/v18/14-428.html.

McMahan, H. B. and Streeter, M. Adaptive bound optimization for online convex optimization. In *Proceedings of the 23rd Annual Conference on Learning Theory (COLT)*, 2010.

McMahan, H. B., Holt, G., Sculley, D., Young, M., Ebner, D., Grady, J., Nie, L., Phillips, T., Davydov, E., Golovin, D., et al. Ad click prediction: a view from the trenches. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1222–1230, 2013.

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

McMahan, H. B., Andrew, G., Erlingsson, U., Chien, S., Mironov, I., Papernot, N., and Kairouz, P. A general approach to adding differential privacy to iterative training procedures. *arXiv preprint arXiv:1812.06210*, 2018.

Mironov, I. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pp. 263–275. IEEE, 2017.

Nasr, M., Song, S., Thakurta, A., Papernot, N., and Carlini, N. Adversary instantiation: Lower bounds for differentially private machine learning. In *IEEE S and P (Oakland)*, 2021.

Overflow, S. The Stack Overflow Data, 2018. https://www.kaggle.com/stackoverflow/stackoverflow.

Papernot, N., Chien, S., Song, S., Thakurta, A., and Erlingsson, U. Making the shoe fit: Architectures, initializations, and tuning for learning with privacy, 2020a. URL https://openreview.net/forum?id=rJg851rYwH.

Papernot, N., Thakurta, A., Song, S., Chien, S., and Erlingsson, Ú. Tempered sigmoid activations for deep learning with differential privacy. *arXiv preprint arXiv:2007.14191*, 2020b.

Pichapati, V., Suresh, A. T., Yu, F. X., Reddi, S. J., and Kumar, S. Adaclip: Adaptive clipping for private sgd. *arXiv preprint arXiv:1908.07643*, 2019.

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

Reddi, S., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečnỳ, J., Kumar, S., and McMahan, H. B. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

Robbins, H. and Monro, S. A stochastic approximation method. *The annals of mathematical statistics*, pp. 400–407, 1951.

Shalev-Shwartz, S. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012.

Shalev-Shwartz, S., Shamir, O., Srebro, N., and Sridharan, K. Stochastic convex optimization. In *COLT 2009 - The 22nd Conference on Learning Theory, Montreal, Quebec, Canada, June 18-21, 2009*, 2009. URL http://www.cs.mcgill.ca/%7Ecolt2009/papers/018.pdf#page=1.

Shalev-Shwartz, S. et al. Online learning and online convex optimization. *Foundations and trends in Machine Learning*, 4(2):107–194, 2011.

Smith, A. and Thakurta, A. (nearly) optimal algorithms for private online learning in full-information and bandit settings. In *Advances in Neural Information Processing Systems*, pp. 2733–2741, 2013.

Song, C. and Shmatikov, V. Auditing data provenance in text-generation models. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 196–206, 2019.

Song, S., Chaudhuri, K., and Sarwate, A. D. Stochastic gradient descent with differentially private updates. In *2013 IEEE Global Conference on Signal and Information Processing*, pp. 245–248. IEEE, 2013.

Thakkar, O., Andrew, G., and McMahan, H. B. Differentially private learning with adaptive clipping. *CoRR*, abs/1905.03871, 2019. URL http://arxiv.org/abs/1905.03871.

Thakkar, O., Ramaswamy, S., Mathews, R., and Beaufays, F. Understanding unintended memorization in federated learning. *arXiv preprint arXiv:2006.07490*, 2020.

Tramèr, F. and Boneh, D. Differentially private learning needs better features (or much more data). In *International Conference on Learning Representations (ICLR)*, 2021.

Wang, Y.-X., Balle, B., and Kasiviswanathan, S. P. Subsampled rényi differential privacy and analytical moments accountant. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1226–1235. PMLR, 2019.

Warner, S. L. Randomized response: A survey technique for eliminating evasive answer bias. *J. of the American Statistical Association*, 60(309):63–69, 1965.

Wu, X., Li, F., Kumar, A., Chaudhuri, K., Jha, S., and Naughton, J. F. Bolt-on differential privacy for scalable stochastic gradient descent-based analytics. In Salihoglu, S., Zhou, W., Chirkova, R., Yang, J., and Suciu, D. (eds.), *Proceedings of the 2017 ACM International Conference on Management of Data, SIGMOD*, 2017.

Xiao, L. Dual averaging methods for regularized stochastic learning and online optimization. *The Journal of Machine Learning Research*, 11:2543–2596, 2010.

Zhu, Y. and Wang, Y.-X. Poission subsampled rényi differential privacy. In *International Conference on Machine Learning*, pp. 7634–7642. PMLR, 2019.