

# Differentially Private Federated Learning: An Information-Theoretic Perspective

Shahab Asoodeh<sup>†</sup>, Wei-Ning Chen<sup>\*</sup>, Flavio P. Calmon<sup>†</sup>, and Ayfer Özgür<sup>\*</sup>

<sup>†</sup>Harvard University, <sup>\*</sup>Stanford University

**Abstract**—We propose a new technique for deriving the differential privacy parameters in federated learning (FL). We consider the setting where a machine learning model is iteratively trained using stochastic gradient descent (SGD) and only the last update is publicly released. In this approach, we interpret each training iteration as a Markov kernel. We then quantify the impact of the kernel on privacy parameters via the contraction coefficient of the  $E_\gamma$ -divergence that underlies differential privacy. To do so, we generalize the well-known Dobrushin’s ergodicity coefficient, originally defined in terms of total variation distance, to a family of  $f$ -divergences. We then analyze the convergence rate of SGD under the proposed private FL framework.

## I. INTRODUCTION

Federated Learning (FL) [1] is a distributed method for training machine learning models. In the prototypical setting, users compute gradients on their local data and send them to a server referred to as the *central aggregator* (uplink update). The local gradients are then aggregated into an update by the server, which is then sent back to users (down-link update). This iterative distributed algorithm has recently gained attention due to its inherit parallelization, storage, and communication efficiency. Although users never share their local data directly during each iteration—only gradients are transmitted—FL can still compromise user privacy [2, 3].

In this paper, we derive privacy guarantees for FL. We adopt differential privacy (DP) as our privacy metric of choice, since DP has become the standard for large-scale model fitting (e.g., [4–14]). We make two key assumptions. First, we assume that users communicate over encrypted channels with a *trusted* aggregator. Second, we assume that the aggregator releases the model parameters publicly only after a certain number of iterations and hides all intermediate updates. Augenstein et al. [15] recently studied the same setting where, after  $T$  iterations, the last model parameters are used to generate synthetic data for data inspection purposes. This assumption is also in line with the recent works [16, 17] where the privacy amplification resulting from hiding intermediate updates was quantified. However, these works differ from ours in that we allow for subsampling of users and adopt the *approximate* DP as the measure of privacy. In contrast, in [16, 17] noise at each iteration is the only source of randomness (i.e., no

subsampling) and the privacy was given in terms of Rényi differential privacy.<sup>1</sup>

To characterize the privacy-utility trade-off, we analyze the convergence rate of stochastic gradient descent (SGD) under the proposed privacy-preserving FL framework. We consider two common data generation scenarios. First, we let each local sample be generated i.i.d. according to an unknown source  $P_X$ . In this case, we show that the convergence rate is degraded by an additive term  $C_0\sigma^2/n$ , where  $\sigma$  is the variance of the noise added in each iteration. Second, we consider heterogeneous data and make no assumption on the underlying distribution. Due to the one-pass nature of the proposed FL algorithm, the standard SGD analysis fails in this regime since the local gradient obtained at each step is no longer unbiased. To overcome this, we generalize the results on without-replacement SGD [20], proving a similar upper bound on the convergence rate. Our results specify the relation between convergence rate, noise level as well as sample and batch size. Moreover, it sheds light on how to select these hyper-parameters to achieve better privacy-utility trade-off.

**Notation.** For any set  $A$ , we denote by  $\mathcal{P}(A)$  the set of all probability distributions on  $A$ . Given two sets  $\mathcal{Y}$  and  $\mathcal{Z}$ , a Markov kernel (i.e., channel)  $K$  is a mapping from  $\mathcal{Y}$  to  $\mathcal{P}(\mathcal{Z})$  given by  $y \mapsto K(y)$ . Given  $\mu \in \mathcal{P}(\mathcal{Y})$  and a Markov kernel  $K : \mathcal{Y} \rightarrow \mathcal{P}(\mathcal{Z})$ , we let  $\mu K$  denote the output distribution of  $K$  when the input distribution is  $\mu$ , i.e.,  $\mu K = \int K(y)\mu(dy)$ .

## II. PRELIMINARIES

### A. Differential Privacy

Let  $\mathcal{X}^n$  be the set of all possible datasets of size  $n$ , where each entry takes values in  $\mathcal{X}$ . A pair of datasets  $x \in \mathcal{X}^n$  and  $x' \in \mathcal{X}^n$  are neighboring (denoted by  $x \sim x'$ ) if they differ in exactly one entry. A randomized mechanism  $\mathcal{M}$  acts on each  $x \in \mathcal{X}^n$  and generates a random variable with distribution  $\mathcal{M}_x$ . A mechanism  $\mathcal{M}$  is said to be  $(\varepsilon, \delta)$ -DP [21], for  $\varepsilon \geq 0$  and  $\delta \in [0, 1]$ , if we have

$$\sup_{x \sim x'} \sup_A [\mathcal{M}_x(A) - e^\varepsilon \mathcal{M}_{x'}(A)] \leq \delta, \quad (1)$$

where the first supremum is taken over all measurable sets  $A$ .

<sup>1</sup>We note that Rényi differential privacy can be converted into  $(\varepsilon, \delta)$ -DP, according to [18]. However, as shown in [19], the resulting privacy guarantees are weaker than what would be obtained by directly computing  $\varepsilon$  and  $\delta$ .

This work was supported in part by NSF under grants CIF 1900750, CIF CAREER 1845852, and CCF 0939370.

## B. $f$ -Divergences

Given a convex function  $f : [0, \infty) \rightarrow \mathbb{R}$  with  $f(1) = 0$ , the  $f$ -divergence [22, 23] between two probability measures  $\mu$  and  $\nu$  is defined as  $D_f(\mu\|\nu) := \mathbb{E}_\nu \left[ f\left(\frac{d\mu}{d\nu}\right) \right]$ . This includes several popular measures: KL-divergence,  $\chi^2$ -divergence, and total variation distance TV are  $f$ -divergences for  $f(t) = t \log(t)$ ,  $f(t) = (t - 1)^2$ , and  $f(t) = \frac{1}{2}|t - 1|$ , respectively.

Given  $\varepsilon \geq 0$ , consider the convex function  $E_\varepsilon(t) := (t - e^\varepsilon)_+$ , where  $(a)_+ := \max\{0, a\}$ . The corresponding  $f$ -divergence, denoted by  $E_\varepsilon(P\|Q)$ , is called  $E_\varepsilon$ -divergence (or sometimes *hockey-stick divergence* [24]) and is explicitly defined as

$$E_\varepsilon(\mu\|\nu) = \int_{\mathcal{Y}} (d(\mu - e^\varepsilon \nu)(y))_+. \quad (2)$$

From the Neyman-Pearson lemma we can obtain an alternative formula for  $E_\varepsilon(\mu\|\nu)$  as  $E_\varepsilon(\mu\|\nu) = \sup_A [\mu(A) - e^\varepsilon \nu(A)]$ , implying that the DP constraint (1) can be equivalently expressed in terms of  $E_\varepsilon$ -divergence [25]:  $\mathcal{M}$  is  $(\varepsilon, \delta)$ -DP if and only if

$$\sup_{x \sim x'} E_\varepsilon(\mathcal{M}_x\|\mathcal{M}_{x'}) \leq \delta. \quad (3)$$

This  $E_\varepsilon$ -divergence representation of DP was used in [26–30] to prove new privacy results or simplify the proofs of existing results. The following properties of  $E_\varepsilon$ -divergence can be readily proved:

- $0 \leq E_\varepsilon(\mu\|\nu) \leq \text{TV}(\mu, \nu)$  for any  $\varepsilon > 0$ . The upper bound is equality if and only if  $\varepsilon = 0$ ,
- $\varepsilon \mapsto E_\varepsilon(\mu\|\nu)$  is continuous and strictly decreasing on  $(0, \text{TV}(\mu, \nu)]$ ,
- $(\mu, \nu) \mapsto E_\varepsilon(\mu\|\nu)$  is convex,
- $E_\varepsilon(\mu\|\nu)$  decreases by post-processing (Data-Processing Inequality). That is,  $E_\varepsilon(\mu K\|\nu K) \leq E_\varepsilon(\mu\|\nu)$  for any Markov kernel (or a channel)  $K$ .

The Data Processing Inequality can be strict for non-trivial kernels. This can be captured by the *contraction coefficient* [31]  $\eta_\varepsilon(K)$  of  $K$  under  $E_\varepsilon$ -divergence as

$$\eta_\varepsilon(K) := \sup_{\substack{\mu, \nu: \\ E_\varepsilon(\mu\|\nu) \neq 0}} \frac{E_\varepsilon(\mu K\|\nu K)}{E_\varepsilon(\mu\|\nu)}. \quad (4)$$

This quantity has been studied in detail in [19]. In particular, it was shown that  $\eta_\varepsilon(K)$  enjoys a remarkably simple two-point characterization.

**Theorem 1** ([19]). *For any  $\varepsilon \geq 0$  and  $K : \mathcal{Y} \rightarrow \mathcal{P}(\mathcal{Z})$ , we have*

$$\eta_\varepsilon(K) = \sup_{y_1, y_2 \in \mathcal{Y}} E_\varepsilon(K(y_1)\|K(y_2)). \quad (5)$$

When  $\varepsilon = 0$ , this theorem reduces to the well-known Dobrushin's theorem [32] that has been an instrumental result in several statistical problems, see, e.g., [32–35].

In this paper, we are concerned with the Gaussian Markov kernel specified by  $K(y) = \mathcal{N}(y, \sigma^2 I)$  for some  $y \in \mathbb{R}^d$  and  $\sigma > 0$ . To compute the contraction coefficient of such kernels, we need the following lemma, whose proof is essentially the same as [26, Lemma 6].

**Lemma 1.** *For  $m_1, m_2 \in \mathbb{R}^d$  and  $\sigma > 0$ , we have*

$$E_\varepsilon(\mathcal{N}(m_1, \sigma^2 I)\|\mathcal{N}(m_2, \sigma^2 I)) = \theta_\varepsilon\left(\frac{\|m_1 - m_2\|}{\sigma}\right),$$

where  $\theta_\varepsilon : [0, \infty) \rightarrow [0, 1]$  is given by

$$\theta_\varepsilon(r) := Q\left(\frac{\varepsilon}{r} - \frac{r}{2}\right) - e^\varepsilon Q\left(\frac{\varepsilon}{r} + \frac{r}{2}\right), \quad (6)$$

and  $Q(t) = \frac{1}{\sqrt{2\pi}} \int_t^\infty e^{-u^2/2} du$ .

In light of Theorem 1 and Lemma 1, it follows that Gaussian kernels have a trivial contraction coefficient, i.e.,  $\eta_\varepsilon(K) = 1$  (for instance by choosing  $m_1 = 0$  and  $m_2$  with  $\|m\|_2 \rightarrow \infty$ ). However, if the input is assumed to be restricted to a bounded subset of  $\mathbb{R}^d$ , then  $\eta_\varepsilon(K) < 1$ .

**Lemma 2** ([19]). *Let  $\mathcal{Y} \subset \mathbb{R}^d$  be a bounded set. For the Markov kernel specified by  $K(y) = \mathcal{N}(y, \sigma^2 I)$  for  $y \in \mathcal{Y}$  and  $\sigma > 0$ , we have*

$$\eta_\varepsilon(K) = \theta_\varepsilon\left(\frac{\|\mathcal{Y}\|}{\sigma}\right),$$

where  $\|\mathcal{Y}\| := \max_{y_1, y_2 \in \mathcal{Y}} \|y_1 - y_2\|$ .

The constraint that the input of Gaussian kernels must be bounded is not restrictive in machine learning and is satisfied in many practical algorithms. For instance, each iteration of the *projected* noisy stochastic gradient descent with Gaussian noise (see e.g., [4, 5, 7, 8, 11, 29]) can be viewed as a Gaussian kernel whose input (and output) are values from a compact set. Such kernels are called *projected* Gaussian kernels. We focus on this particular kernel in the next section.

## III. FEDERATED LEARNING

In our federated learning model,  $n$  distributed users send their updates of a shared model to a *trusted* aggregator. At each iteration,  $m$  number of users are chosen uniformly *without replacement*. Each selected user computes a local update, randomizes it via a Gaussian kernel, and returns it to the aggregator. The aggregator adds all the local updates, projects it onto  $\ell_2$ -ball of fixed radius  $\rho$  and then sends the global update back to users. For simplicity, we assume  $m = qn$  and since the subsampling is performed without replacement, the total number of iterations is  $T = \frac{n}{m} = \frac{1}{q}$ . This procedure is described in Algorithm 1. The model we investigate differs from the typical settings studied in the literature in that here the aggregator is expected to publicly display the model parameters only after the  $T$ th iteration. This model is conceptually similar to the recent work of Augenstein et al. [15] where the final model parameters were used to generate synthetic data for the purpose of data inspection under privacy constraints.

### A. Warm-Up: Batches of Size 1

Suppose  $n$  users, each with local data  $x_i$ ,  $i \in [n] := \{1, \dots, n\}$ , communicate over an encrypted channel with the trusted aggregator and send their local updates *one at a time*, i.e.,  $m = 1$ . Although this setting may not be practical, it

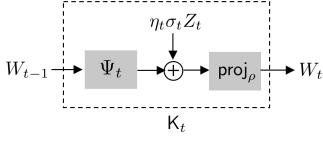


Fig. 1. Iteration  $t$  can be viewed as a Markov kernel that is composed of  $\Psi_t$  defined in (7), Gaussian noise addition and then projection operator onto  $\text{ball}(\rho)$ .

elucidates the proof technique employed for the general setting (i.e.,  $m \geq 1$ ).

Let  $\pi \in \mathcal{S}_n$  be a random permutation map and  $\mathcal{S}_n$  the symmetric group on  $[n]$ . The federated learning algorithm iterates as follows:

- The aggregator samples the initial parameter  $W_0$  in  $\text{ball}(\rho)$ , the  $\ell_2$  ball of radius  $\rho$  in  $\mathbb{R}^d$ , according to a distribution  $\mu_0$  and sends it to user  $\pi(1)$ .
- User  $\pi(1)$  uses  $W_0$  and her local data  $x_{\pi(1)}$  to compute the update  $\tilde{W}_1 := \eta_1 \nabla \ell(W_0, x_{\pi(1)}) + \eta_1 \sigma_1 Z_1$ , where  $Z_1 \sim \mathcal{N}(0, \mathbf{I})$ . This update is then sent back to the aggregator.
- Upon receipt of  $\tilde{W}_1$ , the aggregator computes  $W_1 = \text{proj}_\rho(W_0 - \tilde{W}_1)$ , where  $\text{proj}_\rho(\cdot)$  denotes the projection operator onto  $\text{ball}(\rho)$ . Then  $W_1$  is sent to user  $\pi(2)$ .
- Repeat the above procedure until all  $n$  users have sent their updates to the aggregator (i.e.,  $T = n$  iterations). The aggregator releases  $W_T$ .

To obtain the privacy guarantee of this algorithm, we model each iteration as a projected Gaussian Markov kernel. Let  $K_t$  be the Markov kernel associated with the map  $w \mapsto \text{proj}_\rho(\Psi_t(w) - \eta_t \sigma_t Z_t)$  for  $t \in [T]$ , where

$$\Psi_t(w) := w - \eta_t \nabla \ell(w, x_{\pi(t)}), \quad (7)$$

and  $Z_t$  is a random vector sampled from  $\mathcal{N}(0, \mathbf{I})$ . More precisely,  $K_t(w) = \text{proj}_\rho(\mathcal{N}(\Psi_t(w), \eta_t^2 \sigma_t^2 \mathbf{I}))$ . It is clear from Lemma 2 that  $\eta_\varepsilon(K_t) < 1$  for all  $\varepsilon \geq 0$  and  $\rho < \infty$ . Notice that the  $t$ th iteration can be equivalently expressed by  $K_t$  whose input is  $W_{t-1}$  and output is  $W_t$  (see Fig 1). Letting  $\mu_{t-1}$  denote the distribution of  $W_{t-1}$ , we therefore have  $W_t \sim \mu_{t-1} K_t$ .

Now consider a pair of neighboring datasets  $x$  and  $x'$  that differ in the  $i$ th entry (i.e.,  $x_i \neq x'_i$  and  $x_j = x'_j$  for  $j \in [n] \setminus \{i\}$ ) and let  $\mu_t$  and  $\mu'_t$  be the distributions of the  $W_t$  when algorithm runs on  $x$  and  $x'$ , respectively. Let  $t = \pi^{-1}(i)$  (or equivalently  $\pi(t) = i$ ). Clearly,  $\mu_j = \mu'_j$  for all  $j \in [t-1]$ . Also,  $\mu_t = \mu_{t-1} K_t$  and  $\mu'_t = \mu_{t-1} K'_t$  where  $K'_t$  is the Markov kernel associated with the map  $w \mapsto \text{proj}_\rho(\Psi'_t(w) - \eta_t \sigma_t Z_t)$  and

$$\Psi'_t(w) := w - \eta_t \nabla \ell(w, x'_i).$$

It follows, according to (3), that the algorithm is  $(\varepsilon, \delta)$ -DP if  $\mathsf{E}_\varepsilon(\mu_T \| \mu'_T) \leq \delta$ , for all  $i \in [n]$ . By the definition of  $\eta_\varepsilon$ , we have

$$\mathsf{E}_\varepsilon(\mu_T \| \mu'_T) \leq \mathsf{E}_\varepsilon(\mu_{T-1} \| \mu'_{T-1}) \eta_\varepsilon(K_T)$$

---

**Algorithm 1** Federated learning with a trusted aggregator

---

- 1: **Input:** Dataset  $\{x_1, \dots, x_n\} \in \mathbb{R}^{nd}$ , learning rate  $\{\eta_t\}$ , batch size  $m$ , noise variances  $\{\sigma_t^2\}$ , initial distribution  $\mu_0$
  - 2: Choose  $W_0 \sim \mu_0$
  - 3: **for**  $t = 1$  **to**  $T$  **do**
  - 4:   Take batch  $B_t \subset [n]$  of size  $m$  uniformly without replacement
  - 5:   **Local update:**  $W_{t-1}^j = \eta_t [\nabla \ell(W_{t-1}, x_j) + \sigma_t Z_t^j]$ ,  $\forall j \in B_t$  and  $Z_t^j \sim \mathcal{N}(0, \mathbf{I})$
  - 6:   **Upload:**  $W_{t-1}^j$  is sent to aggregator
  - 7:   **Model aggregation:** aggregator updates the model parameter as  $W_t = \text{proj}_\rho(W_{t-1} - \frac{1}{m} \sum_{j \in B_t} W_{t-1}^j)$
  - 8: **end for**
  - 9: **Output:**  $W_T$
- 

$$\leq \mathsf{E}_\varepsilon(\mu_{T-2} \| \mu'_{T-2}) \eta_\varepsilon(K_T) \eta_\varepsilon(K_{T-1}).$$

Applying this for  $T - t$  times, we obtain

$$\begin{aligned} \mathsf{E}_\varepsilon(\mu_T \| \mu'_T) &\leq \mathsf{E}_\varepsilon(\mu_t \| \mu'_t) \prod_{j=t+1}^T \eta_\varepsilon(K_j) \\ &= \mathsf{E}_\varepsilon(\mu_{t-1} K_t \| \mu_{t-1} K'_t) \prod_{j=t+1}^T \eta_\varepsilon(K_j) \end{aligned} \quad (8)$$

Consequently, the computation of  $\delta$  boils down to computing the contraction coefficient of projected Gaussian kernels and  $\mathsf{E}_\varepsilon$ -divergence between mixture of projected Gaussian distributions with the same variance. The former can be bounded directly by Lemma 2. To obtain an upper bound for the latter, we apply Jensen's inequality (recall that  $(\mu, \nu) \mapsto \mathsf{E}_\varepsilon(\mu \| \nu)$  is convex), Data Processing Inequality (to get rid of the projection operator), and Lemma 1. When  $m > 1$ , we can proceed similarly except that the resulting Markov kernel at each iteration is composed of  $m$  projected Gaussian kernels.

### B. Batch of size $m$

Here we assume that at each iteration, the aggregator shares the global update with  $m$  users. In this setting,  $T = \frac{n}{m}$  and, in lieu of permutation, we define a mapping which assigns each  $i \in [n]$  to a single batch.

**Theorem 2.** Let the loss function  $w \mapsto \ell(w, x)$  be convex,  $L$ -Lipschitz and  $\beta$ -smooth for all  $x \in \mathcal{X}$  and also  $\eta \leq \frac{2}{\beta}$ . Then Algorithm 1 is  $(\varepsilon, \delta)$ -DP for  $\varepsilon \geq 0$  and

$$\delta = \frac{m}{n} \sum_{t=1}^T \theta_\varepsilon \left( \frac{2L}{\sqrt{m} \sigma_t} \right) \prod_{j=t+1}^T \theta_\varepsilon \left( \frac{2\rho\sqrt{m}}{\eta_j \sigma_j} \right),$$

where  $\theta_\varepsilon$  is defined in (6). In particular, if  $\eta_t = \eta$  and  $\sigma_t = \sigma$  for all  $t \in [T]$ , we have

$$\delta = \frac{m}{n} \theta_\varepsilon \left( \frac{2L}{\sqrt{m} \sigma} \right) \frac{1 - \theta_\varepsilon \left( \frac{2\rho\sqrt{m}}{\eta \sigma} \right)^{\frac{n}{m}}}{1 - \theta_\varepsilon \left( \frac{2\rho\sqrt{m}}{\eta \sigma} \right)}.$$

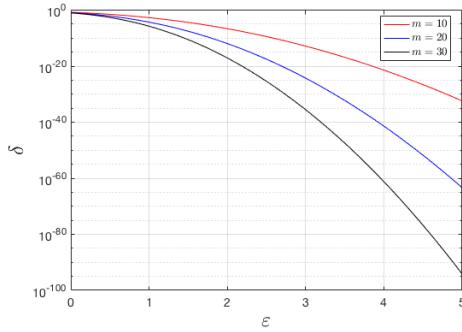


Fig. 2. Differential privacy parameters of Algorithm 1 for different sub-sampling rates according to Theorem 2. The parameters of algorithm are as follows:  $\eta = 0.5$ ,  $L = 1$ ,  $\rho = 1$ ,  $\sigma = 1.5$ ,  $n = 100$ .

The proof of this theorem (and other results) are given in [36]. Note that the convexity and smoothness of  $\ell(\cdot, x)$  are used in the proof of Theorem 2 only to obtain an upper bound for  $\|\Psi_t(\text{ball}(\rho))\|$ . This was shown via standard results in convex analysis (e.g., Prop 18 in [16]) that state  $w \mapsto w - \eta\nabla\ell(w, x)$  is contractive for  $\eta \leq \frac{2}{\beta}$  if  $\ell(\cdot, x)$  is convex,  $L$ -Lipschitz, and  $\beta$ -smooth, implying  $\|\Psi_t(\text{ball}(\rho))\| \leq 2\rho$ . However, one can easily show that in the absence of convexity and smoothness,  $\|\Psi_t(\text{ball}(\rho))\| \leq 2(\rho + \eta_t L)$ . Therefore, the convexity and smoothness can be relaxed in Theorem 2 at the cost of slightly looser bound. If, however, the cost function is strongly convex, then Theorem 2 can be improved as it is known that  $w \mapsto w - \eta\nabla\ell(w, x)$  is contractive with Lipschitz constant strictly smaller than 1 (see, e.g., [37, Theorem 3.12]). In Fig. 2, we demonstrate the privacy parameters obtained from Theorem 2 for  $\eta_t = 0.5$ ,  $\sigma_t = 1.5$ , and different sub-sampling rates  $q = 0.1, 0.2, 0.3$ . As illustrated in this figure, the more users are involved in each iteration, the better the privacy guarantee is.

#### IV. PRIVACY-UTILITY TRADE-OFF

In this section, we apply the technique developed in the last section to study the convergence rate of private SGD (Algorithm 1). In particular, we consider two canonical data generation scenarios.

- Distributional SGD (stochastic optimization): each local data  $X_i$  is drawn identically and independently from an unknown source  $P_X$ , and the goal is to minimize

$$F(W) \triangleq \mathbb{E}_{P_X} [\ell(W, X)] + r(W), \quad (9)$$

for some loss function  $\ell(\cdot)$  and regularization  $r(\cdot)$ .

- Distribution-free SGD: each  $x_i \in \mathcal{X}$ , and the goal is to minimize

$$F(W) \triangleq \frac{1}{n} \sum_{i=1}^n \ell(W, x_i) + r(W). \quad (10)$$

Note that in the standard SGD setting, the sever observes a local unbiased estimate of gradient vector  $\nabla F(W_t)$  at each

iteration  $t$  and updates the global model  $W_t$  accordingly, thus as long as we select user uniformly at random, there is no difference between distributional and distribution-free SGD. However, due to the privacy constraint, in Algorithm 1 each user is selected randomly but *without-replacement*, hence the updates need not be (conditionally) unbiased, rendering the traditional convergence analysis of SGD futile.

#### A. Distributional SGD

By applying standard SGD convergence results (for instance Theorem 1 in [38]), we obtain the following utility guarantee.

**Corollary 1.** Suppose  $\mathcal{W} \subseteq \text{ball}(\rho)$  and  $F(W) \triangleq \mathbb{E}_{P_X} [\ell(W, X)]$  is  $\lambda$  strongly convex and  $\beta$  smooth on  $\mathcal{W}$ , with  $\|\nabla F(W)\|_2^2 \leq D^2$  and  $\text{Var}_{P_X} (\nabla \ell(W, X)) \leq G^2$ . Let  $T \triangleq \frac{n}{m}$  and  $W_T$  be the output of Algorithm 1. Then by choosing  $\eta_t = \frac{1}{\lambda t}$  and  $\sigma_t = \sigma$ , we have

$$\begin{aligned} \mathbb{E}[F(W_T)] - \inf_{W \in \mathcal{W}} F(W) &\leq \frac{2\beta \left( D^2 + \frac{G^2 + \sigma^2}{m} \right)}{\lambda^2 T} \\ &= \frac{2\beta (mD^2 + G^2 + \sigma^2)}{\lambda^2 n}. \end{aligned}$$

Moreover, by Theorem 2, Algorithm 1 satisfies  $(\varepsilon, \delta)$ -DP with

$$\begin{aligned} \delta &= \frac{m}{n} \sum_{t=1}^T \theta_\varepsilon \left( \frac{2L}{\sqrt{m}\sigma} \right) \prod_{j=t+1}^T \theta_\varepsilon \left( \frac{2\rho\sqrt{\lambda m j}}{\sigma} \right) \\ &\leq \frac{m}{n} \theta_\varepsilon \left( \frac{2L}{\sqrt{m}\sigma} \right) \frac{1 - \theta_\varepsilon \left( \frac{2\rho\lambda n}{\sqrt{m}\sigma} \right)^{\frac{n}{m}}}{1 - \theta_\varepsilon \left( \frac{2\rho\lambda}{\sqrt{m}\sigma} \right)}, \end{aligned} \quad (11)$$

where the inequality is due to the monotonicity of  $r \mapsto \theta_\varepsilon(r)$ . Therefore we see that the price of privacy is an additive term  $\sigma^2/n$  in the convergence rate. Notice that a straightforward upper bound on (11) is  $\theta_\varepsilon \left( \frac{2L}{\sqrt{m}\sigma} \right)$ , making clear that stronger privacy guarantee can be obtained by increasing either noise level  $\sigma$  or batch size  $m$  (see Fig 2).

#### B. Distribution-free SGD

In general, the local data at each user is highly heterogeneous (e.g., a mobile phone) and, thus, better represented by the distribution-free setting. However, since in Algorithm 1 each user is selected without replacement at each iteration, the resulting local gradient vector is no longer an unbiased estimate of the global gradient, making the traditional SGD convergence analysis fail. Nevertheless, borrowing the idea from [20], we show that sampling each user without replacement does no harm on the convergence rate compared to the classic SGD (i.e. with-replacement SGD).

- a) *Utility guarantee:* We start with the following convergence result.

**Corollary 2.** Suppose  $\mathcal{W} \subseteq \text{ball}(\rho)$  and that  $F(\cdot) \triangleq \frac{1}{n} \sum_i f_i(\cdot)$  is  $\lambda$  strongly convex on  $\mathcal{W}$ . Assume  $f_i(W) = \ell(\langle W, x_i \rangle) + r(W)$  where  $\|x_i\| \leq 1$ ,  $r(\cdot)$  is possibly some regularization term, and  $\ell$  is  $L$ -Lipschitz and  $\beta$ -smooth

on  $\{z : z = \langle W, x \rangle, W \in \mathcal{W}, \|x\| \leq 1\}$ . Furthermore, suppose  $\sup_{W \in \mathcal{W}} \|\nabla f_i(W)\| \leq G$ . Then choose  $\eta_t = \frac{1}{\lambda t}$ ,  $m = 1$  and let  $W_t$  be the model after  $t$ -th round in Algorithm 1, we have (for a universal constant  $c$ )

$$\begin{aligned} & \mathbb{E} \left[ \frac{1}{n} \sum_{t=1}^n F(W_t) \right] - \inf_{W \in \mathcal{W}} F(W) \\ & \leq c \frac{\left( (L + \mu B)^2 + G^2 \right) \log(T)}{\lambda n} + \frac{\sum_t \eta_t \sigma_t^2}{n}. \end{aligned}$$

**Remark 1.** Note that Theorem 2 is essentially the result of Theorem 3 in [20], except that now we replace the update rule  $W_{t+1} = \text{Proj}_\rho(W_t - \eta_t \nabla f_{\pi(t)}(W_t))$  with  $W_{t+1} = \text{Proj}_\rho(W_t - \eta_t (\nabla f_{\pi(t)}(W_t) + \sigma_t Z_t))$ .

To extend Corollary 2 to batch-size  $m$ , simply rewrite

$$F(\cdot) = \frac{1}{n} \sum_i f_i(\cdot) = \frac{1}{T} \sum_{t=1}^T \left( \frac{1}{m} \sum_{i \in B_t} f_i(\cdot) \right) \triangleq \frac{1}{T} \sum_{t=1}^T g_t(\cdot),$$

where  $T \triangleq \frac{n}{m}$  and  $B_t$  is a random size- $m$  batch selected without replacement. Then the update rule in Algorithm 1 can be viewed as

$$W_{t+1} = \text{Proj}_\rho \left( W_t - \eta_t \left( \nabla g_{\pi(t)}(W_t) + \frac{1}{m} \sum_{j \in B_t} \sigma_t Z_t^j \right) \right),$$

and applying Corollary 2 yields

$$\begin{aligned} & \mathbb{E} \left[ \frac{1}{T} \sum_{t=1}^T F(W_t) \right] - \inf_{W \in \mathcal{W}} F(W) \\ & \leq c \frac{\left( (L + \mu B)^2 + G^2 \right) \log(T)}{\lambda T} + \frac{\sum_t \eta_t \sigma_t^2}{m T} \\ & = c \frac{m \left( (L + \mu B)^2 + G^2 \right) \log(n/m)}{\lambda n} + \frac{\sum_t \eta_t \sigma_t^2}{n}. \end{aligned}$$

b) *Privacy guarantee:* Corollary 2 only ensures the convergence of  $\frac{1}{T} \sum_{t=1}^T F(W_t)$  instead of the output  $W_T$ . Notice that if we replace the output of Algorithm 1 with  $\bar{W} \triangleq \frac{1}{T} \sum_{t=1}^T W_t$ , the privacy guarantee in Theorem 2 need not hold. To address this issue, we consider a *randomly stopped* version of Algorithm 1 as in [19], where after running  $\tau \sim \text{uniform}(T)$  rounds of update, we stop and return  $W_\tau$ . In this case, the output satisfies  $\mathbb{E}[W_\tau] = \frac{1}{T} \sum_{t=1}^T W_t$ , hence the convergence result in Corollary 2 holds.

Motivated by [19, Theorem 5], we give the following privacy guarantee for the randomly stopped version of Algorithm 1:

**Corollary 3.** Let  $T \triangleq \frac{n}{m}$  and  $\tau \sim \text{uniform}(T)$ . If we run Algorithm 1 for  $\tau$  rounds and return  $W_\tau$ , then  $W_\tau$  satisfies  $(\varepsilon, \delta)$ -DP with

$$\delta = \frac{1}{T^2} \sum_{\tau=1}^T \sum_{t=1}^{\tau} \theta_\varepsilon \left( \frac{2L}{\sqrt{m}\sigma_t} \right) \prod_{j=t+1}^{\tau} \theta_\varepsilon \left( \frac{2\rho\sqrt{m}}{\eta_j \sigma_j} \right).$$

Moreover, if  $\eta^* \triangleq \min_{t \in [T]} \eta_t$  and  $\sigma^* \triangleq \min_{t \in [T]} \sigma_t$ , then we

can also pick  $\delta$  as

$$\delta = \frac{1}{T^2} \theta_\varepsilon \left( \frac{2L}{\sqrt{m}\sigma^*} \right) \sum_{\tau=1}^T \frac{1 - \theta_\varepsilon^\tau \left( \frac{2\rho\sqrt{m}}{\eta^* \sigma^*} \right)}{1 - \theta_\varepsilon \left( \frac{2\rho\sqrt{m}}{\eta^* \sigma^*} \right)}. \quad (12)$$

For the parameters given in Corollary 2, we have  $\eta^* = \frac{1}{\lambda T}$ . Thus, by choosing  $\sigma_t^2 = \sqrt{n}$ , we obtain the privacy guarantee from (12)

$$\begin{aligned} \delta &= \frac{1}{T^2} \theta_\varepsilon \left( \frac{2L}{\sqrt{m}\sigma^*} \right) \sum_{\tau=1}^T \frac{1 - \theta_\varepsilon^\tau \left( \frac{2\lambda\rho m}{\sqrt{m}\sigma^*} \right)}{1 - \theta_\varepsilon \left( \frac{2\lambda\rho m}{\sqrt{m}\sigma^*} \right)} \\ &\leq \frac{1}{T^2} \theta_\varepsilon \left( \frac{2L}{\sqrt{m}\sqrt[4]{n}} \right) \sum_{\tau=1}^T \frac{1 - \theta_\varepsilon^\tau \left( \frac{2\lambda\rho m^{\frac{3}{4}}}{\sqrt{m}} \right)}{1 - \theta_\varepsilon \left( \frac{2\lambda\rho m^{\frac{3}{4}}}{\sqrt{m}} \right)}, \end{aligned}$$

and the convergence rate from Corollary 2 as  $\tilde{O} \left( \frac{1}{\sqrt{n}} \vee \frac{m}{n} \right)$ , where  $\vee$  denotes the maximization operator.

We close this section with a few remarks. In Corollary 1 and Corollary 2, we assume the loss function to be strongly convex (which generally holds if we add a regularization term  $r(W)$ ). One can remove this assumption, as in standard SGD convergence analysis (e.g. Chapter 14 in [39]), and obtain  $O \left( \frac{1}{\sqrt{T}} \right)$  rate (instead of  $\tilde{O} \left( \frac{1}{T} \right)$ ). Moreover, comparing Corollary 1 with Corollary 2, we see that having i.i.d. property on local samples has two consequences: 1) it is no longer needed to randomly stop Algorithm 1, and 2) the loss function  $\ell(\cdot, x_i)$  does not need to take the form  $\ell(\langle \cdot, x_i \rangle)$ . Finally, the randomly stopped version of Algorithm 1 can be replaced with  $\alpha$ -suffix averaging [38], that is, the stopping time  $\tau$  is chosen  $\tau \sim \text{uniform}(\alpha T : T)$  for some  $\alpha \in (0, 1)$ . This can potentially improve the privacy guarantee (12) in Corollary 3 by a constant factor.

## V. CONCLUSION

In this work, we introduce a new approach for computing differential privacy (DP) parameters via contraction coefficient of Markov kernels under a certain  $f$ -divergence, namely  $\mathsf{E}_\varepsilon$ -divergence. In this approach, we interpret federated learning algorithms as a composition of several Markov kernels and express the DP privacy parameters as the product of contraction coefficients of such kernels. The main assumption is that the algorithm releases the model update only after a certain number of iterations are passed; thus no composition theorems are required. The proof technique relies on a technical theorem that establishes a close-form expression for the contraction coefficient of general Markov kernels under  $\mathsf{E}_\varepsilon$ -divergence.

This approach can be adapted to study the more typical scenario where the model updates get released after each iteration. The privacy analysis in this case amounts to deriving the contraction coefficient of a Markov kernel that is obtained by tensor product of all  $T$  kernels, i.e., a kernel with  $T$ -tuple input and output, under  $\mathsf{E}_\varepsilon$ -divergence.

## REFERENCES

- [1] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *AISTATS*, 2016.
- [2] M. Fredrikson, S. Jha, and T. Ristenpart, “Model inversion attacks that exploit confidence information and basic countermeasures,” in *Proc. Conf. Computer and Communications Security (CCS)*, p. 13221333, 2015.
- [3] L. Melis, C. Song, E. D. Cristofaro, and V. Shmatikov, “Inference attacks against collaborative learning,” *ArXiv*, vol. abs/1805.04049, 2018.
- [4] X. Wu, F. Li, A. Kumar, K. Chaudhuri, S. Jha, and J. Naughton, “Bolt-on differential privacy for scalable stochastic gradient descent-based analytics,” in *SIGMOD*, pp. 1307–1322, 2017.
- [5] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, “Differentially private empirical risk minimization,” *Journal of Machine Learning Research*, vol. 12, no. Mar, pp. 1069–1109, 2011.
- [6] K. Chaudhuri and N. Mishra, “When random sampling preserves privacy,” in *Advances in Cryptology - CRYPTO 2006*, pp. 198–213, Springer Berlin Heidelberg, 2006.
- [7] R. Bassily, A. Smith, and A. Thakurta, “Private empirical risk minimization: Efficient algorithms and tight error bounds,” in *Proc. Symp. Foundations of Computer Science, FOCS 14*, pp. 464–473, 2014.
- [8] R. Bassily, V. Feldman, K. Talwar, and A. Guha Thakurta, “Private stochastic convex optimization with optimal rates,” in *Neural Inf. Proc. Systems*, pp. 11282–11291, 2019.
- [9] P. Jain, P. Kothari, and A. Thakurta, “Differentially private online learning,” in *Proc. Conf. Learning Theory*, vol. 23, pp. 24.1–24.34, 25–27 Jun 2012.
- [10] A. G. Thakurta and A. Smith, “Differentially private feature selection via stability arguments, and the robustness of the lasso,” in *Proc. Conf. Learning Theory*, vol. 30, pp. 819–850, 12–14 Jun 2013.
- [11] S. Song, K. Chaudhuri, and A. D. Sarwate, “Stochastic gradient descent with differentially private updates,” in *IEEE Global Conf. Signal and Inf. Proc.*, pp. 245–248, 2013.
- [12] P. Jain and A. G. Thakurta, “(near) dimension independent risk bounds for differentially private learning,” in *Proc. Int. Conf. Machine Learning*, vol. 32, pp. 476–484, 22–24 Jun 2014.
- [13] A. Smith, A. Thakurta, and J. Upadhyay, “Is interaction necessary for distributed private learning?,” in *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 58–77, 2017.
- [14] D. Wang, M. Ye, and J. Xu, “Differentially private empirical risk minimization revisited: Faster and more general,” in *Proc. Neural Inf. Proc. Systems*, p. 27192728, 2017.
- [15] S. Augenstein, H. B. McMahan, D. Ramage, S. Ramaswamy, P. Kairouz, M. Chen, R. Mathews, and B. A. y Arcas, “Generative models for effective ml on private, decentralized datasets,” in *International Conference on Learning Representations*, 2020.
- [16] V. Feldman, I. Mironov, K. Talwar, and A. Thakurta, “Privacy amplification by iteration,” *FOCS*, pp. 521–532, 2018.
- [17] B. Balle, G. Barthe, M. Gaboardi, and J. Geumlek, “Privacy amplification by mixing and diffusion mechanisms,” in *NeurIPS*, pp. 13277–13287, 2019.
- [18] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, “Deep learning with differential privacy,” in *Proc. ACM SIGSAC CCS*, pp. 308–318, 2016.
- [19] S. Asoodeh, M. Diaz, and F. P. Calmon, “Privacy analysis of online learning algorithms via contraction coefficients,” *arXiv 2012.11035*, 2020.
- [20] O. Shamir, “Without-replacement sampling for stochastic gradient methods,” *Advances in neural information processing systems*, vol. 29, pp. 46–54, 2016.
- [21] C. Dwork, F. McSherry, K. Nissim, and A. Smith, “Calibrating noise to sensitivity in private data analysis,” in *Proc. Theory of Cryptography (TCC)*, pp. 265–284, 2006.
- [22] S. M. Ali and S. D. Silvey, “A general class of coefficients of divergence of one distribution from another,” *Journal of Royal Statistics*, vol. 28, pp. 131–142, 1966.
- [23] I. Csiszár, “Information-type measures of difference of probability distributions and indirect observations,” *Studia Sci. Math. Hungar.*, vol. 2, pp. 299–318, 1967.
- [24] N. Sharma and N. A. Warsi, “Fundamental bound on the reliability of quantum information transmission,” *CoRR*, vol. abs/1302.5281, 2013.
- [25] G. Barthe and F. Olmedo, “Beyond differential privacy: Composition theorems and relational logic for  $f$ -divergences between probabilistic programs,” in *Proc. ICALP*, pp. 49–60, 2013.
- [26] B. Balle and Y.-X. Wang, “Improving the Gaussian mechanism for differential privacy: Analytical calibration and optimal denoising,” in *ICML*, vol. 80, pp. 394–403, 10–15 July 2018.
- [27] Y.-X. Wang, B. Balle, and S. P. Kasiviswanathan, “Subsampled Rényi differential privacy and analytical moments accountant,” in *AISTAT*, vol. 89, pp. 1226–1235, 16–18 Apr 2018.
- [28] B. Balle, G. Barthe, and M. Gaboardi, “Privacy amplification by subsampling: Tight analyses via couplings and divergences,” in *NeurIPS*, pp. 6280–6290, 2018.
- [29] B. Balle, G. Barthe, M. Gaboardi, J. Hsu, and T. Sato, “Hypothesis testing interpretations and Rényi differential privacy,” in *AISTAT*, 2020.
- [30] B. Balle, J. Bell, A. Gascón, and K. Nissim, “The privacy blanket of the shuffle model,” vol. 11693, pp. 638–667, 2019.
- [31] R. Ahlswede and P. Gács, “Spreading of sets in product spaces and hypercontraction of the markov operator,” *Ann. Probab.*, vol. 4, pp. 925–939, 12 1976.
- [32] R. L. Dobrushin, “Central limit theorem for nonstationary markov chains. I,” *Theory Probab. Appl.*, pp. 65–80, 1956.
- [33] A. Kontorovich and M. Raginsky, “Concentration of measure without independence: A unified approach via the martingale method,” in *Convexity and Concentration*, pp. 183–210, Springer New York, 2017.
- [34] Y. Polyanskiy and Y. Wu, “Dissipation of information in channels with input constraints,” *IEEE Trans. Inf. Theory*, vol. 62, pp. 35–55, Jan 2016.
- [35] A. Xu and M. Raginsky, “Converses for distributed estimation via strong data processing inequalities,” in *IEEE Int. Sympos. Inf. Theory (ISIT)*, pp. 2376–2380, 2015.
- [36] S. Asoodeh, W.-N. Chen, F. P. Calmon, and A. Özgür, “Differentially private federated learning: An information-theoretic perspective,” <https://scholar.harvard.edu/files/shahab/files/isit21fl.pdf>, 2021.
- [37] S. Bubeck, *Convex Optimization: Algorithms and Complexity*, vol. 8. Foundations and Trends in Machine Learning, 2015.
- [38] A. Rakhlin, O. Shamir, and K. Sridharan, “Making gradient descent optimal for strongly convex stochastic optimization,” *arXiv preprint arXiv:1109.5647*, 2011.
- [39] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning - From Theory to Algorithms*. Cambridge University Press, 2014.