Differentially Private Federated Learning: An Information-Theoretic Perspective

Shahab Asoodeh†, Wei-Ning Chen*, Flavio P. Calmon†, and Ayfer Özgür*
†Harvard University, *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 via the contraction coefficient of the $\epsilon$-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 (downlink 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 it is the measure of privacy we of choice. In contrast, in [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 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.1

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 and 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 $P \in \mathcal{P}(\mathcal{Y})$ and a Markov kernel $K : \mathcal{Y} \rightarrow \mathcal{P}(\mathcal{Z})$, we let $PK$ denote the output distribution of $K$ when the input distribution is $P$, i.e., $PK = \int K(y)P(dy)$.

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

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

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

B. Information Theory

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 $D_f(\mu||\nu) := \mathbb{E}_\nu [f(\frac{\mu}{\nu})]$. This includes several popular measures: KL-divergence, $\chi^2$-divergence, and total

1We 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 $\delta$. 
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 \( f_\varepsilon(t) := (t - e^\varepsilon)_+ \), 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
\[
E_\varepsilon(\mu\|\nu) := \int_Y (d(\mu - e^{\varepsilon}\nu)(y) +). \tag{2}
\]

This divergence appeared in [25] for proving converse channel coding results. 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 [26]: \( M \) is \( (\varepsilon, \delta) \)-DP if and only if
\[
\sup_{x \sim x'} E_\varepsilon(M_x\|M_{x'}) \leq \delta. \tag{3}
\]

This \( E_\varepsilon \)-divergence representation of DP was used in [27–31] to prove new privacy results or simplify the proofs of existing results.

The following properties of \( E_\varepsilon \)-divergence can be readily proved:

- \( 0 \leq E_\varepsilon(\mu\|\nu) \leq TV(\mu, \nu) \) 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 \((0, TV(\mu, \nu)]\).
- \( (\mu, \nu) \mapsto E_\varepsilon(\mu\|\nu) \) 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) \( K \).

Data Processing inequality is typically strict for non-trivial kernels. To account for this, it is customary to consider the contraction coefficient [32] \( \eta_\varepsilon(K) \) of \( K \) under \( E_\varepsilon \)-divergence as
\[
\eta_\varepsilon(K) := \sup_{\mu, \nu, \varepsilon} \frac{E_\varepsilon(\mu K\|\nu K)}{E_\varepsilon(\mu\|\nu)}. \tag{4}
\]

This quantity has been recently in details 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
\[
\eta_\varepsilon(K) = \sup_{y_1, y_2 \in \mathcal{Y}} E_\varepsilon(K(y_1)\|K(y_2)). \tag{5}
\]

When \( \varepsilon = 0 \), this theorem reduces to the well-known Dobrushin’s theorem [33] that has been an instrumental result in several statistical problems, see, e.g. [33–36].

In this paper, we are concerned with the Gaussian Markov kernel specified by \( K(y) = \mathcal{N}(y, \sigma^2I) \) 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 [27, Lemma 6].

**Lemma 1.** For \( m_1, m_2 \in \mathbb{R}^d \) and \( \sigma > 0 \), we have
\[
E_\varepsilon(\mathcal{N}(m_1, \sigma^2I)\|\mathcal{N}(m_2, \sigma^2I)) = \theta_\varepsilon\left( \frac{\|m_1 - m_2\|}{\sigma} \right),
\]
where \( \theta_\varepsilon : [0, \infty) \rightarrow [0, 1] \) is given by
\[
\theta_\varepsilon(\varepsilon) := Q\left( \frac{\varepsilon - r}{2} \right) - e^\varepsilon Q\left( \frac{\varepsilon + r}{2} \right),
\]
and \( Q(t) := \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-u^2/2} du \).

In light of Theorem 1 and Lemma 1, it follows that Gaussian kernels has a trivial contraction coefficient, i.e., \( \eta_\varepsilon(K) = 1 \) (for instance by choosing \( m_1 = 0 \) and \( m_2 \) with \( \|m_2\|_2 \rightarrow \infty \)). 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 \( K(y) = \mathcal{N}(y, \sigma^2I) \) 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, 30]) 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. Then, each of users selected computes a local update, randomizes it via a Gaussian kernel, and then sends it to the aggregator. The aggregator aggregates all these local updates, projects it onto \( \ell^2 \)-ball of fixed radius \( \rho \) and then sends the global update back to users. For notational simplicity, we assume \( m = qn \) and since the subsampling is performed without replacement, the total number of iteration 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 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 the synthetic data for the purpose of data inspection under privacy constraint.

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

Suppose \( n \) users, each with local data \( x_i, i \in [n] := \{1, \ldots, n\} \), are to communicate over an encrypted communication channel to a trusted party and send their local update to shared model one at a time, i.e., \( m = 1 \). Although this
setting may not be practical, it illuminates the proof technique employed for the general setting (i.e., \( m \geq 1 \)).

Let \( \pi \in S_n \) be a random permutation map and \( S_n \) is 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 \( \mathbb{E}_\rho \) 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 \( \hat{W}_1 := \eta_1 \nabla \ell(W_0, x_{\pi(1)}) + \eta_1 \sigma_1 Z_1 \), where \( Z_1 \sim \mathcal{N}(0, I) \). This update is then sent back to the aggregator.
- Upon receipt of \( \hat{W}_1 \), the aggregator computes \( W_1 = \text{proj}_\rho(W_0 - \hat{W}_1) \), where \( \text{proj}_\rho(\cdot) \) denotes the projection operator onto \( \text{ball}(\rho) \). Then \( W_1 \) is sent to user \( \pi(2) \).
- Continue the above procedure until all \( n \) users send the aggregator their updates (i.e., \( T = n \) is the number of 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

\[
\Psi_t(w) := w - \eta_t \nabla \ell(w, x_{\pi(t)}),
\]

and \( Z_t \) is a random vector sampled from \( \mathcal{N}(0, I) \). More precisely, \( K_t(w) = \text{proj}_\rho(\mathcal{N}(\Psi_t(w), \eta_t^2 \sigma_t^2 I)) \). It is clear from Lemma 2 that \( \eta_t(K_t) < 1 \) for all \( \epsilon \geq 0 \) 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 \( W_t \sim \mu_{t-1} K_t \).

![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 ball(\( \rho \)).](image)

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 \( w \mapsto \text{proj}_\rho(\Psi_t'(w) - \eta_t \sigma_t Z_t) \),

where

\[
\Psi_t'(w) := w - \eta_t \nabla \ell(w, x'_i).
\]

In light of (3), one concludes the algorithm is \((\epsilon, \delta, \beta)\)-DP if \( E_\epsilon(\mu_T \| \mu'_T) \leq \delta \), for all \( i \in [n] \). By definition, we have

\[
E_\epsilon(\mu_T \| \mu'_T) \leq E_\epsilon(\mu_{T-1} \| \mu'_{T-1}) \eta_\epsilon(K_T).
\]

Algorithm 1: Federated learning with a trusted aggregator

1: **Input:** Dataset \( \{x_1, \ldots, x_n\} \in \mathbb{R}^{nd} \), learning rate \( \{\eta_t\} \), batch size \( m \), noise variance \( \{\sigma_t^2\} \), 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:** \( W_{t+1}^j = \eta_t \left[ \nabla \ell(W_t - x_j) + \sigma_t Z_t^j \right] \), \( \forall j \in B_t \) and \( Z_t^j \sim \mathcal{N}(0, I) \)
6: **Upload:** \( W_{t-1} \) is sent to aggregator
7: **Model aggregation:** aggregator updates the model parameter as \( W_t = \text{proj}_\rho(W_{t-1} - \frac{1}{m} \sum_{j \in B_t} W_j^j) \)
8: end for
9: **Output:** \( W_T \)

\[
\leq E_\epsilon(\mu_{T-2} \| \mu'_{T-2}) \eta_\epsilon(K_T) \eta_\epsilon(K_{T-1}).
\]

Applying this for \( T - t \) times, we obtain

\[
E_\epsilon(\mu_T \| \mu'_T) \leq E_\epsilon(\mu_t \| \mu'_t) \prod_{j=t+1}^T \eta_\epsilon(K_j) = E_\epsilon(\mu_{t-1} K_t \| \mu_{t-1} K'_t) \prod_{j=t+1}^T \eta_\epsilon(K_j) \quad (8)
\]

Consequently, the computation of \( \delta \) boils down to computing the contraction coefficient of projected Gaussian kernels and \( E_\epsilon \)-divergence between mixture of projected Gaussian distributions with the same variance. The former can be tackled via Lemma 2. The latter, however, involves Jensen’s inequality (recall that \((\mu, \nu) \mapsto E_\epsilon(\mu \| \nu) \) is convex), the data processing inequality (to get rid of the projection operator) and Lemma 1. We will elaborate further in the next section where we prove the main result.

B. Batch of size \( m \)

Here we assume 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 \( x \in X \) and also \( \eta \leq \frac{2}{\beta} \). Then Algorithm 1 is \((\epsilon, \delta)\)-DP for \( \epsilon \geq 0 \) and

\[
\delta = \frac{m \epsilon}{n} \sum_{t=1}^T \prod_{j=t+1}^T \theta_\epsilon \left( \frac{2L}{\sqrt{m} \sigma_t} \right) \left( \frac{\eta t \sqrt{m}}{\eta_\epsilon \sigma_t} \right),
\]

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

\[
\delta = \frac{m \epsilon}{n} \theta_\epsilon \left( \frac{2L}{\sqrt{m} \sigma} \right) \left( \frac{1 - \theta_\epsilon}{1 - \theta_\epsilon \frac{2\sqrt{m} \eta}{\eta_\epsilon \sigma}} \right) \frac{n}{m}.
\]

The proof of this theorem (and other results) are give in [37]. 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 \( \eta \leq \frac{\rho}{2} \) if \( \ell(\cdot, x) \) is convex, \( L \)-Lipschitz, and \( \beta \)-smooth; thus \( \|\Psi_t(\text{ball}(\rho))\| \leq 2\rho \). However, one can easily show that in the absence of convexity and smoothness, \( \|\Psi_t(\text{ball}(\rho))\| \leq 2(\rho + \eta L) \). Therefore, the convexity and smoothness can be relaxed in Theorem 2 at the price of looser and more tedious bound. On the other hand, if the cost function is strongly convex, then Theorem 3.12 in \([39]\)) 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 higher the privacy guarantee is.

IV. PRIVACY-UTILITY TRADE-OFF

In this section, we apply the technique in Section III to study the convergence rate of private SGD (Algorithm 1), which is the main utility function we concerned. In particular, we consider two canonical data generation scenarios.

- **Distributional SGD** (stochastic optimization): each local sample \( X_i \) is drawn identically and independently from an unknown source \( P_X \), and the goal is to minimize
  \[
  F(W) \triangleq \mathbb{E}_{P_X}[\ell(W, X)] + r(W),
  \]
  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
  \[
  F(W) \triangleq \frac{1}{n} \sum_{i=1}^{n} \ell(W, X_i) + r(W).
  \]

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, so as long as we select user uniformly at random \( i_t \overset{i.i.d.}{\sim} \text{uniform}(n) \), there is no difference between distributional or distribution-free SGD. However, due to the privacy constraint, in Algorithm 1 each user \( i \) is picked randomly but without-replacement, so the updates are no longer (conditional) unbiased, making the traditional analysis on SGD failed.

A. Distributional SGD

By applying standard SGD convergence results (for instance Theorem 1 in \([39]\)), we obtain the following utility guarantee:

**Corollary 1.** Suppose \( \mathcal{W} \subseteq \mathcal{B}(\ell_2, \rho) \) and that \( F(W) \triangleq \mathbb{E}_{P_X}[\ell(W, X)] \) 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 \( T \triangleq \frac{m}{\sigma} \) and \( W_T \) be the output of Algorithm 1. Then by choosing \( \eta_t = \frac{1}{T} \) and \( \sigma_t = \sigma \), we have

\[
\mathbb{E}[F(W_T)] - \inf_{W \in \mathcal{W}} F(W) \leq \frac{2\beta (D^2 + G^2 + \sigma^2)}{\lambda^2 T} = \frac{2\beta (mD^2 + G^2 + \sigma^2)}{\lambda^2 n}.
\]

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

\[
\delta = \frac{m}{n} \sum_{t=1}^{T} \theta_{\varepsilon} \left( \frac{2L}{\sqrt{m\sigma}} \right) \prod_{t=1}^{T} \theta_{\varepsilon} \left( \frac{2\sqrt{\lambda mn}}{\sigma} \right) \\
\leq \frac{m}{n} \theta_{\varepsilon} \left( \frac{2L}{\sqrt{m\sigma}} \right) \left( 1 - \theta_{\varepsilon} \left( \frac{2\lambda m}{\sqrt{m\sigma}} \right) \right) \xrightarrow{\triangleq} \frac{\varepsilon}{m},
\]

where the inequality is due to \( \eta_t \leq \eta_T \) and 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 \( \theta_{\varepsilon} \left( \frac{2L}{\sqrt{m\sigma}} \right) \), so this implies one can get stronger privacy guarantee by increasing either noise level \( \sigma \) or batch size \( m \).

B. Distribution-free SGD

In general, the local data at each local device is typically highly heterogeneous, so the distribution-free setting captures the feature of federate learning better. 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).

\textit{a) Utility guarantee:} We start with the following convergence result:

**Corollary 2.** Suppose \( \mathcal{W} \subseteq \mathcal{B}(\ell_2, \rho) \) and that \( F(\cdot) \triangleq \frac{1}{n} \sum_{i=1}^{n} f_i(\cdot) \) is \( \lambda \) strongly convex on \( \mathcal{W} \). Assume \( f_i(W) = \ell((W, x_i)) + r(W) \) where \( |x_i| \leq 1 \), \( r(\cdot) \) is possibly some regularization term, and \( \ell \) is \( L \)-Lipschitz and \( \beta \)-smooth on \( \{z : z = (W, x), W \in \mathcal{W}, |x| \leq 1\} \). Furthermore, suppose \( \sup_{W \in \mathcal{W}} \|\nabla f_i(W)\| \leq G \). Then choose \( \eta_t = \frac{1}{T}, m = 1 \) and
let $W_t$ be the model after $t$-th round in Algorithm 1, we have (for a universal constant $c$)

$$
\mathbb{E} \left[ \frac{1}{n} \sum_{t=1}^{n} F(W_t) \right] - \inf_{W \in \mathcal{W}} F(W) 
\leq c \left( \frac{(L + \mu B)^2}{\lambda n} + \frac{G^2}{\lambda n} + \frac{1}{n} \sum_{i} \eta_i \sigma_i^2 \right).
$$

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

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

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

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

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

and applying Corollary 2 yields

$$
\mathbb{E} \left[ \frac{1}{T} \sum_{t=1}^{T} F(W_t) \right] - \inf_{W \in \mathcal{W}} F(W) 
\leq c \left( \frac{(L + \mu B)^2}{\lambda n} + \frac{G^2}{\lambda n} + \frac{1}{n} \sum_{i} \eta_i \sigma_i^2 \right) 
\leq c \left( \frac{(L + \mu B)^2}{\lambda n} + \frac{G^2}{\lambda n} + \frac{1}{n} \sum_{i} \eta_i \sigma_i^2 \right).
$$

**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 will no longer 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 $W_\tau$. This case, the output satisfies $\mathbb{E}[W_\tau] = \frac{1}{T} \sum_{t=1}^{T} W_t$, so the convergence result in Corollary 2 holds.

Motivated by the [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 $(\epsilon, \delta)$-DP with

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

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

$$
\delta = \frac{1}{T^2} \theta_e \left( \frac{2L}{\sqrt{m} \sigma^*} \right) \sum_{t = 1}^{T} \frac{1 - \theta_e \left( \frac{2\rho \sqrt{m}}{\eta^* \sigma^*} \right)}{1 - \theta_e \left( \frac{2\rho \sqrt{m}}{\eta^* \sigma^*} \right)}.
$$

For the parameters given in Corollary 2, we have $\eta^* = \frac{1}{\lambda T}$, so if we pick $\sigma^2 = \sqrt{n}$ and plug into (12), we obtain

$$
\delta = \frac{1}{T^2} \theta_e \left( \frac{2L}{\sqrt{m \lambda T}} \right) \sum_{t = 1}^{T} \frac{1 - \theta_e \left( \frac{2\rho \sqrt{m}}{\sqrt{m \lambda T}} \right)}{1 - \theta_e \left( \frac{2\rho \sqrt{m}}{\sqrt{m \lambda T}} \right)}
\leq \frac{1}{T^2} \theta_e \left( \frac{2L}{\sqrt{m \lambda T}} \right) \sum_{t = 1}^{T} \frac{1 - \theta_e \left( \frac{2\rho \sqrt{m}}{\sqrt{m \lambda T}} \right)}{1 - \theta_e \left( \frac{2\rho \sqrt{m}}{\sqrt{m \lambda T}} \right)}.
$$

and by Corollary 2, the convergence rate is $\tilde{O} \left( \frac{1}{\sqrt{n} \sqrt{\frac{m}{T}}} \right)$.

We close this section with a few remarks in order. In Corollary 1 and Corollary 2, we assume the loss function to be strongly convex (which generally holds if we add a regularization term $\tau(W)$). One can remove this assumption, as in standard SGD convergence analysis (e.g. Chapter 14 in [40]), and obtain $O \left( \frac{1}{T^2} \right)$ rate (instead of $\tilde{O} \left( \frac{1}{T} \right)$). Secondly, comparing Corollary 1 with Corollary 2, we see that the advantage of having i.i.d. property on local samples includes 1) we no longer need to randomly stop Algorithm 1 to obtain the averaging $\frac{1}{T} \sum_{t=1}^{T} F(W_t)$, and 2) the loss function $\ell(\cdot, x_i)$ does not need to take the form $\ell(\cdot, x_i)$. Finally, the randomly stopped version of Algorithm 1 can be replaced with $\alpha$-suffix averaging [39], 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 $E_\epsilon$-divergence. In this approach, we interpret federated learning algorithm 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 $E_\epsilon$-divergence.

This approach can be adapted to study the more typical scenario where the model updates are 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 $E_\epsilon$-divergence.
Appendix

Proof of Theorem 1. This result was originally proved in [19]. Nevertheless, we provide the proof here for the convenience.

Given two probability measures $\mu$ and $\nu$ defined on $\mathcal{Y}$, define $\phi(y) = (\mu(y) - e^{\nu(y)})_+$ and $\phi'(y) = (-\mu(y) + e^{\nu(y)})_+$ for any $y \in \mathcal{Y}$. Note that since $\frac{1}{2}\|\mu - e^{\nu(y)}\| = E_\epsilon(\|\mu\| + \frac{1}{2}(e^\epsilon - 1)$ and $\|\phi\|_1 = E_\epsilon(\|\nu\| + e^{\epsilon} - 1$. Letting $E_\epsilon$ denote $E_\epsilon(\|\mu\|)$ for brevity, we can write

$$
\|\mu K - e^{\nu K}\|_1 = \|\mu(dy) - e^{\nu(dy)}\|K(y)
$$

$$
= \int_{\mathcal{Y}} \int_{\mathcal{X}} \phi(dy)K(y) - \int_{\mathcal{Y}} \phi'(dy)K(y)
$$

$$
= \int_{\mathcal{Y}} \|\phi\| \int_{\mathcal{X}} \phi'(dy)K(y)
$$

$$
= \int_{\mathcal{Y}} \|\phi\| \int_{\mathcal{X}} \phi'(dy)K(y)
$$

$$
- \|\phi\| \int_{\mathcal{Y}} \phi'(dy)K(y)
$$

$$
\leq \max_{x,x'} \int_{\mathcal{Y}} \|\phi\| K(y) - \|\phi\| K(y)
$$

$$
\leq \max_{x,x'} \int_{\mathcal{Y}} K(y) - (x + e^{\epsilon} - 1)K(y)
$$

Notice that it follows from the definition of $E_\epsilon$-divergence that

$$
E_\epsilon(\mu K||\nu K) = \frac{1}{2} \int |d(\mu K - e^{\nu K})| - \frac{1}{2}(e^{\epsilon} - 1).
$$

Consequently, we obtain from (14) that

$$
E_\epsilon(\mu K||\nu K) \leq E_\epsilon(\mu||\nu) \max_{y,y'} E_\epsilon(K(y)||y').
$$

and hence $\eta_\epsilon(K) \leq \max_{y,y'} E_\epsilon(K(y)||y')$. Now we show that this inequality is indeed an equality. Fix $y_1 \neq y_2 \in \mathcal{Y}$ and $\delta \in (0,1)$. Define $\delta = \delta(y_1,y_2)$ and $\nu_\delta = (1 - \delta \epsilon)(\delta) + (1 - \delta e^{-\epsilon})\eta_\epsilon(\delta)(y_2)$, where $\delta := 1 - \delta, y_1 \notin \{y_1, y_2\}$ and $\epsilon_\delta$ is the indicator function. It is easy to verify that $E_\delta(\mu K||\nu_\delta)$ is also $\delta$. We also have $\delta K = \delta (y_1) + \delta K(y_2)$ and $\nu_\delta = (1 - \delta e^{-\epsilon})\eta_\epsilon(y_2)$. Hence, by (2),

$$
E_\delta(\mu K||\nu K) = \delta \int_{\mathcal{Y}} \int_{\mathcal{X}} \epsilon_\delta(K(y_1) - e^{\nu K(y_2)})(y_2)
$$

$$
\leq \delta E_\epsilon(K(y_1)||K(y_2)),
$$

where $\epsilon := \log(1 + e^{\epsilon} - 1)$. Therefore, we obtain that

$$
\eta_\epsilon(K) \geq \max_{y,y'} E_\epsilon(K(y)||y') = E_\epsilon(K(y)||y_2).
$$

By continuity of $\epsilon \mapsto E_\epsilon(\mu ||\nu)$, we obtain from above

$$
\eta_\epsilon(K) \geq \lim_{\epsilon \rightarrow 0} E_\epsilon(K(y)||y_2) = E_\epsilon(K(y)||y_2).
$$

Since $y_1$ and $y_2$ are arbitrary, the desired result follows.

Proof of Theorem 2. Consider two neighboring datasets $x = \{x_1, \ldots, x_n\}$ and $\tilde{x}' = \{x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n\}$. Let $\mu_t$ and $\mu'_t$ be the distribution of $W_t$ the output of the tth iteration when running on $x$ and $x'$, respectively. To derive $\delta$ for any given $\epsilon$, we need to compute $E_\epsilon(\mu_t ||\mu'_t)$. Let $\pi : [n] \mapsto [T]$ specifies an assignment of users to each batch, i.e., $\pi(i) = t$ if $i \in B_t$. Note that $\mu_t = \mu'_t t < \pi(i)$. We now identify each iteration with a projected Markov kernel. At iteration $t$, the aggregator generates

$$
W_t = \text{proj}_t(W_{t-1} - \frac{\eta}{m} \sum_{j \in B_t} \nabla \ell(W_{t-1}, x_j) - \delta_t Z_t),
$$

where $Z_t$ is now standard Gaussian random variable and $\delta_t^2 := \frac{\eta^2 \sigma^2}{m}$. Hence, iteration $t$ can be realized by $K_t$ a projected Markov kernel associated with the mapping $w \mapsto \text{proj}_t(\Psi_t(w) - \delta_t Z_t)$ where

$$
\Psi_t(w) = w - \frac{\eta}{m} \sum_{j \in B_t} \nabla \ell(w, x_j).
$$

Notice that $K_t$ receives $W_{t-1}$ and generates $W_t$ both taking values in ball($\rho$). Due to the strong data processing inequality (see 8) and convexity of $(\mu, \nu) \mapsto E_\epsilon(\mu||\nu)$ for any $\epsilon \geq 0$, we can write

$$
E_\epsilon(\mu T ||\mu'_T) \leq \sum_{t=1}^T \text{Pr}(\pi(i) = t) E_\epsilon(\mu||\mu'_t) \prod_{j=t+1}^T \eta_t(K_t)
$$

$$
= q \sum_{t=1}^T E_\epsilon(\mu_t ||\mu'_t) \prod_{j=t+1}^T \eta_t(K_t)
$$

(15)

To compute a bound for $\delta_t$, it thus suffices to compute $\eta_t(K_t)$ for $t \in [T]$. $K_t$ is defined for $t \in [T]$ as follows

$$
\eta_t(K_t) = \sup_{w_1, w_2 \in \text{ball}(\mu)} E_\epsilon(K_t(w_1)||K_t(w_2))
$$

$$
\leq \sup_{w_1, w_2 \in \text{ball}(\mu)} E_\epsilon(\psi_t(w_1)||\psi_t(w_2), \delta_t^2)
$$

$$
= \sup_{w_1, w_2} E_\epsilon(\psi_t(w_1)||\psi_t(w_2), \delta_t^2)
$$

$$
= \theta_t(\frac{\psi_t(\text{ball}(\mu))}{\delta_t})
$$

(16)

$$
\leq \theta_t(\frac{\psi_t(\text{ball}(\mu))}{\delta_t})
$$

(17)

$$
\leq \theta_t(\frac{\psi_t(\text{ball}(\mu))}{\delta_t})
$$

(18)

where the equality in (17) follows from Lemma 1 and (6), the inequality in (16) is due to the data processing inequality:

$$
E_\epsilon(\text{proj}_t(\psi_t(w_1), \delta_t^2)) ||\text{proj}_t(\psi_t(w_2), \delta_t^2)
$$

$$
\leq E_\epsilon(\psi_t(w_1)||\psi_t(w_2), \delta_t^2)
$$

$$
\text{proj}_t(\psi_t(w_1), \delta_t^2)) ||\text{proj}_t(\psi_t(w_2), \delta_t^2)
$$

and finally, the inequality in (18) follows from the following two facts: (1) Since the loss functions $w \mapsto \ell(w, x)$ is convex and $\beta$-smooth for all $x \in \mathcal{X}$, then $w \mapsto \ell(w, x)$ is convex for $\eta \leq \frac{\beta}{\delta^2}$ (see e.g., Prop 18 in [16]) and so is $w \mapsto \psi_t(w)$; and (2) The map $r \mapsto \theta_t(r)$ is increasing.

Next, we compute $E_\epsilon(\mu||\mu'_t)$. Note that

$$
\mu_t = \int_{\text{ball}(\mu)} \mu_{t-1}(dy) K_t(y).
$$

Since $\pi(i) = t$, data point $x_i \in B_t$. For this batch, we define

$$
\psi_t(w) := w - \frac{\eta}{m} \nabla \ell(w, x_i) + \sum_{j \in B_t \setminus \{i\}} \nabla \ell(w, x_j),
$$

and the corresponding Markov kernel $K_t$ associated with $w \mapsto \text{proj}_t(\psi_t(w) - \delta_t Z_t)$.
The convexity of $(\mu, \nu) \mapsto E_\varepsilon(\mu \| \nu)$ implies
\[
E_\varepsilon(\mu_t || \mu'_t) \leq E_\varepsilon(K_t(y) \| K'_t(y))_{\mu_t-1}(dy)
\]
\[
\leq E_\varepsilon(N(\Psi_t(y), \sigma_t^2)1) \| N(\Psi'_t(y), \sigma'_t^2)_{\mu_t-1}(dy)
\]
\[
= \theta_t \left( \frac{2L\eta_t}{m\sigma_t} \right)_{\mu_t-1}(dy)
\]
\[
\leq \theta_t \left( \frac{2L\eta_t}{m\sigma_t} \right)
\]
\[
\leq \theta_t \left( \frac{2L}{m\sigma_t} \right)
\]
where (20) follows from Jensen’s inequality, (21) follows from the data processing inequality, (22) follows from Lemma 1, and finally (23) is due to Lemma 2 as follows: for any $y \in \text{ball}(\mu)$
\[
||\Psi_t(y) - \Psi'_t(y)|| = \frac{\eta_t}{m} ||\nabla \ell(y, x_t) - \nabla \ell(y, x'_t)||
\leq \frac{2L\eta_t}{m}
\]
where the inequality is due to the fact that $w \mapsto \ell(w, x)$ is $L$-Lipschitz for all $x \in \mathcal{X}$ and hence $||\nabla \ell(w, x)|| \leq 2L$.

Plugging (19) and (24) into (15), we obtain
\[
E_\varepsilon(\mu_T \| \mu'_T) \leq q \sum_{t=1}^{T} \theta_t \left( \frac{2L\sqrt{m}}{\sigma} \right) \prod_{j=t+1}^{T} \theta_t \left( \frac{2\rho \sqrt{m}}{\eta_j \sigma_j} \right).
\]
Assuming $\eta_t = \eta$ and $\sigma_t = \sigma$ for $t \in [T]$, the above upper-bound can be simplified as
\[
E_\varepsilon(\mu_T \| \mu'_T) \leq q \theta_t \left( \frac{2L\sqrt{m}}{\sigma} \right) \prod_{j=t+1}^{T} \theta_t \left( \frac{2\rho \sqrt{m}}{\eta_j \sigma_j} \right)
\]
\[
= q \theta_t \left( \frac{2L\sqrt{m}}{\sigma} \right) \prod_{j=t+1}^{T} \theta_t \left( \frac{2\rho \sqrt{m}}{\eta_j \sigma_j} \right)
\]
The desired result then follows by invoking (3). \[\blacksquare\]

**Proof of Corollary 2.** Observe that
\[
E \left[ \|W_{t+1}\|_2^2 \right] = E \left[ \left\| \text{Proj}_{\mu}(W_{t} - \eta_t \left( \nabla f_{\sigma(t)}(W_{t}) + \sigma_t Z_{t} \right)) \right\|_2^2 \right]
\leq E \left[ \|W_{t} - \eta_t \left( \nabla f_{\sigma(t)}(W_{t}) + \sigma_t Z_{t} \right) \|_2^2 \right]
\leq E \left[ \|W_{t} - \eta_t \nabla f_{\sigma(t)}(W_{t}) \|_2^2 \right] + \eta_t^2 E \left[ \|\sigma_t Z_{t}\|_2^2 \right]
\leq E \left[ \|W_{t}\|_2^2 \right] - 2\eta_t E \left[ \left( \nabla f_{\sigma(t)}(W_{t}), W_{t} \right) \right] + \eta_t^2 \left( G^2 + \sigma_t^2 \right).
\]
Follow the rest of the proof in Theorem 3 of [20], we obtain the desired result. \[\blacksquare\]

**Proof of Corollary 3.** Following the analysis in Theorem 2, let $\mu_t$ and $\mu'_t$ be the distributions of $W_t$ with two respect to two neighboring datasets $x$ and $x'$ which differ at the $i$-th item. Then we have
\[
E_\varepsilon(\mu_t \| \mu'_t) = E_\varepsilon \left( \frac{1}{T} \sum_{t=1}^{T} \mu_t \| \frac{1}{T} \sum_{t=1}^{T} \mu'_t \right) \leq \frac{1}{T} \sum_{t=1}^{T} E_\varepsilon(\mu_t \| \mu'_t)
\]
\[
\leq \frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{T} \Pr(\pi(i) = t) E_\varepsilon(\mu_t \| \mu'_t) \prod_{j=t+1}^{T} \eta_{\mu}(K_{j})
\]
\[
= \frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{T} E_\varepsilon(\mu_t \| \mu'_t) \prod_{j=t+1}^{T} \eta_{\mu}(K_{j})
\]