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 parameters via the contraction coefficient of the E_γ-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 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 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 σ 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 $y \mapsto \mathsf{K}(y)$. Given $P \in \mathcal{P}(\mathcal{Y})$ and a Markov kernel $\mathsf{K} : \mathcal{Y} \to \mathcal{P}(\mathcal{Z})$, we let $P\mathsf{K}$ denote the output distribution of K when the input distribution is P, i.e., $P\mathsf{K} = \int \mathsf{K}(y)P(\mathrm{d}y)$.

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 (ε, δ)-DP [21], for $\varepsilon \geq 0$ and $\delta \in [0, 1]$, if we have

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

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

B. Information Theory

Given a convex function $f : [0, \infty) \to \mathbb{R}$ with $f(1) = 0$, the f-divergence [22, 23] between two probability measures μ and ν is defined as $D_f(\mu\|\nu) := \mathbb{E}_\nu\left[f\left(\frac{\mathrm{d}\mu}{\mathrm{d}\nu}\right)\right]$. This includes several popular measures: KL-divergence, χ^2-divergence, and total

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

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 $\mathsf{E}_\varepsilon(P\|Q)$, is called E_ε-divergence (or sometimes *hockey-stick divergence* [24]) and is explicitly defined as

$$\mathsf{E}_\varepsilon(\mu\|\nu) = \int_{\mathcal{Y}} \left(\mathrm{d}(\mu - e^\varepsilon\nu)(y)\right)_+. \tag{2}$$

This divergence appeared in [25] for proving converse channel coding results. From the Neyman-Pearson lemma we can obtain an alternative formula for $\mathsf{E}_\varepsilon(\mu\|\nu)$ as $\mathsf{E}_\varepsilon(\mu\|\nu) = \sup_A \left[\mu(A) - e^\varepsilon\nu(A)\right]$, implying that the DP constraint (1) can be equivalently expressed in terms of E_ε-divergence [26]: $\mathcal{M}$ is (ε,δ)-DP if and only if

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

This E_ε-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_ε-divergence can be readily proved:

- $0 \leq \mathsf{E}_\varepsilon(\mu\|\nu) \leq \mathsf{TV}(\mu,\nu)$ for any $\varepsilon > 0$. The upper bound is equality if and only if $\varepsilon = 0$,
- $\varepsilon \mapsto \mathsf{E}_\varepsilon(\mu\|\nu)$ is continuous and strictly decreasing on $(0, \mathsf{TV}(\mu,\nu)]$,
- $(\mu,\nu) \mapsto \mathsf{E}_\varepsilon(\mu\|\nu)$ is convex,
- $\mathsf{E}_\varepsilon(\mu\|\nu)$ decreases by post-processing (data-processing inequality). That is, $\mathsf{E}_\varepsilon(\mu\mathsf{K}\|\nu\mathsf{K}) \leq \mathsf{E}_\varepsilon(\mu\|\nu)$ 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] $\eta_\varepsilon(\mathsf{K})$ of K under E_ε-divergence as

$$\eta_\varepsilon(\mathsf{K}) := \sup_{\substack{\mu,\nu:\\ \mathsf{E}_\varepsilon(\mu\|\nu)\neq 0}} \frac{\mathsf{E}_\varepsilon(\mu\mathsf{K}\|\nu\mathsf{K})}{\mathsf{E}_\varepsilon(\mu\|\nu)}. \tag{4}$$

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

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

$$\eta_\varepsilon(\mathsf{K}) = \sup_{y_1,y_2\in\mathcal{Y}} \mathsf{E}_\varepsilon(\mathsf{K}(y_1)\|\mathsf{K}(y_2)). \tag{5}$$

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 $\mathsf{K}(y) = \mathcal{N}(y, \sigma^2\mathrm{I})$ 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 [27, Lemma 6].

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

$$\mathsf{E}_\varepsilon(\mathcal{N}(m_1,\sigma^2\mathrm{I})\|\mathcal{N}(m_2,\sigma^2\mathrm{I})) = \theta_\varepsilon\Big(\frac{\|m_1 - m_2\|}{\sigma}\Big),$$

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

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

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

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

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

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

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 ℓ_2-ball of fixed radius ρ 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 Tth 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 $\pi \in \mathcal{S}_n$ be a random permutation map and $\mathcal{S}_n$ is the symmetric group on $[n]$. The federated learning algorithm iterates as follows:

- The aggregator samples the initial parameter W_0 in $\mathsf{ball}(\rho)$, the ℓ_2 ball of radius ρ in $\mathbb{R}^d$, according to a distribution μ_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 $Z_1 \sim \mathcal{N}(0, \mathrm{I})$. This update is then sent back to the aggregator.
- Upon receipt of $\tilde{W}_1$, the aggregator computes $W_1 = \mathsf{proj}_\rho(W_0 - \tilde{W}_1)$, where $\mathsf{proj}_\rho(\cdot)$ denotes the projection operator onto $\mathsf{ball}(\rho)$. 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 $w \mapsto \mathsf{proj}_\rho\left(\Psi_t(w) - \eta_t\sigma_t Z_t\right)$ for $t \in [T]$, where

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

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

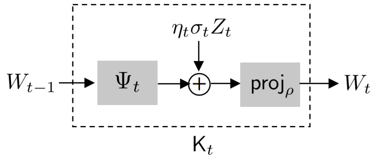

Fig. 1. Iteration t can be viewed as a Markov kernel that is composed of Ψ_t defined in (7), Gaussian noise addition and then projection operator onto $\mathsf{ball}(\rho)$.

Now consider a pair of neighboring datasets x and x' that differ in the ith entry (i.e., $x_i \neq x'_i$ and $x_j = x'_j$ for $j \in [n]\backslash\{i\}$) and let μ_t and μ'_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, $\mu_t = \mu_{t-1}\mathsf{K}_t$ and $\mu'_t = \mu_{t-1}\mathsf{K}'_t$ where K'_t is the Markov kernel associated with the map $w \mapsto \mathsf{proj}_\rho\left(\Psi'_t(w) - \eta_t\sigma_t Z_t\right)$, where

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

In light of (3), one concludes the algorithm is (ε, δ)-DP if $\mathsf{E}_\varepsilon(\mu_T \| \mu'_T) \leq \delta$, for all $i \in [n]$. By definition, we have

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

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

Algorithm 1 Federated learning with a trusted aggregator

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

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

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

Consequently, the computation of δ boils down to computing the contraction coefficient of projected Gaussian kernels and E_ε-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 $(\mu, \nu) \mapsto \mathsf{E}_\varepsilon(\mu \| \nu)$ 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 β-smooth for all $x \in \mathcal{X}$ and also $\eta \leq \frac{2}{\beta}$. Then Algorithm 1 is (ε, δ)-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 θ_ε is defined in (6). In particular, if $\eta_t = \eta$ and $\sigma_t = \sigma$ for all $t \in [T]$, we have

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

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 $\|\Psi_t(\mathsf{ball}(\rho))\|$. 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 β-smooth; thus $\|\Psi_t(\mathsf{ball}(\rho))\| \leq 2\rho$. However, one can easily show that in the absence of convexity and smoothness, $\|\Psi_t(\mathsf{ball}(\rho))\| \leq 2(\rho + \eta_t L)$. 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 2 can be improved as, in this case, $w \mapsto w - \eta\nabla\ell(w,x)$ is contractive with Lipschitz constant strictly smaller than 1 (see, e.g., Theorem 3.12 in [38]). 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 higher the privacy guarantee is.

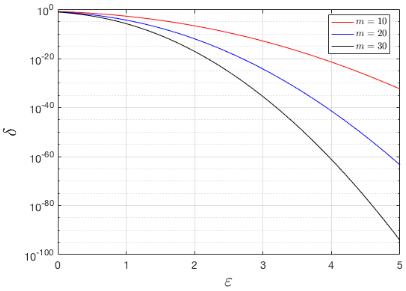

Fig. 2. Differential privacy parameters of Algorithm 1 for different sub-sampling rates according to Theorem 2. The parameters of algorithm are as follows: $\eta = 0.5, L = 1, \rho = 1, \sigma = 1.5, n = 100$.

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
$$F(W) \triangleq \mathbb{E}_{P_X}\left[\ell(W,X)\right] + r(W), \tag{9}$$
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
$$F(W) \triangleq \frac{1}{n}\sum_{i=1}^{n}\ell(W,X_i) + r(W). \tag{10}$$

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, so as long as we select user uniformly at random $i_t \overset{\text{i.i.d.}}{\sim} \mathsf{uniform}(n)$, 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 $F(W) \triangleq \mathbb{E}_{P_X}\left[\ell(W,X)\right]$ is λ strongly convex and β smooth on $\mathcal{W}$, with $\|\nabla F(W)\|_2^2 \leq D^2$ and $\mathsf{Var}_{P_X}\left(\nabla\ell(W,X)\right) \leq G^2$. 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*
$$\begin{aligned}\mathbb{E}\left[F(W_T)\right] - \inf_{W\in\mathcal{W}} F(W) &\leq \frac{2\beta\left(D^2 + \frac{G^2+\sigma^2}{m}\right)}{\lambda^2 T}\\ &= \frac{2\beta\left(mD^2 + G^2 + \sigma^2\right)}{\lambda^2 n}.\end{aligned}$$

Moreover, by Theorem 2, Algorithm 1 satisfies (ε,δ)-DP with
$$\begin{aligned}\delta &= \frac{m}{n}\sum_{t=1}^{T}\theta_\varepsilon\left(\frac{2L}{\sqrt{m}\sigma}\right)\prod_{j=t+1}^{T}\theta_\varepsilon\left(\frac{2\rho\sqrt{\lambda m j}}{\sigma}\right)\\ &\leq \frac{m}{n}\theta_\varepsilon\left(\frac{2L}{\sqrt{m}\sigma}\right)\frac{1-\theta_\varepsilon\left(\frac{2\rho\lambda n}{\sqrt{m}\sigma}\right)^{\frac{n}{m}}}{1-\theta_\varepsilon\left(\frac{2\rho n\lambda}{\sqrt{m}\sigma}\right)},\end{aligned} \tag{11}$$
where the inequality is due to $\eta_j \leq \eta_T$ and the monotonicity of $r \mapsto \theta_\varepsilon(r)$. Therefore we see that the price of privacy is an additive term σ^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 σ 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).

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

Corollary 2. *Suppose $\mathcal{W} \subseteq \mathcal{B}(\ell_2,\rho)$ and that $F(\cdot) \triangleq \frac{1}{n}\sum_i f_i(\cdot)$ is λ strongly convex on $\mathcal{W}$. Assume $f_i(W) = \ell\left(\langle W,x_i\rangle\right) + r(W)$ where $\|x_i\| \leq 1$, $r(\cdot)$ is possibly some regularization term, and ℓ is L-Lipschitz and β-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 $W_{t+1}=\mathsf{Proj}_\rho\left(W_t-\eta_t\nabla f_{\sigma(t)}\left(W_t\right)\right)$ with $W_{t+1}=\mathsf{Proj}_\rho\left(W_t-\eta_t\left(\nabla f_{\sigma(t)}\left(W_t\right)+\sigma_t Z_t\right)\right)$.*

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

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

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

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

and applying Corollary 2 yields

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

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 will no longer hold. To address this issue, we consider a *randomly stopped* version of Algorithm 1 as in [19], where after running $\tau\sim\mathsf{uniform}(T)$ rounds of update, we stop and return W_τ. I this case, the output satisfies $\mathbb{E}\left[W_\tau\right]=\frac{1}{T}\sum_{t=1}^{T}W_t$, 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 $\tau\sim\mathsf{uniform}(T)$. If we run Algorithm 1 for τ rounds and return W_τ, then W_τ satisfies (ε,δ)-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 δ as

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

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

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

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

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 $r(W)$). One can remove this assumption, as in standard SGD convergence analysis (e.g. Chapter 14 in [40]), and obtain $O\left(\frac{1}{\sqrt{T}}\right)$ 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 $\frac{1}{T}\sum_t F(W_t)$, 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 α-suffix averaging [39], that is, the stopping time τ is chosen $\tau\sim\mathsf{uniform}\left(\alpha T:T\right)$ 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_ε-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_ε-divergence.

This approach can be adapted to study the 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 E_ε-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] Y. Polyanskiy, H. V. Poor, and S. Verdú, “Channel coding rate in the finite blocklength regime,” *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2307–2359, 2010.

[26] 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.

[27] 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.

[28] 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.

[29] B. Balle, G. Barthe, and M. Gaboardi, “Privacy amplification by subsampling: Tight analyses via couplings and divergences,” in *NeurIPS*, pp. 6280–6290, 2018.

[30] B. Balle, G. Barthe, M. Gaboardi, J. Hsu, and T. Sato, “Hypothesis testing interpretations and Rényi differential privacy,” in *AISTAT*, 2020.

[31] B. Balle, J. Bell, A. Gascón, and K. Nissim, “The privacy blanket of the shuffle model,” vol. 11693, pp. 638–667, 2019.

[32] 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.

[33] R. L. Dobrushin, “Central limit theorem for nonstationary markov chains. I,” *Theory Probab. Appl.*, pp. 65–80, 1956.

[34] 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.

[35] Y. Polyanskiy and Y. Wu, “Dissipation of information in channels with input constraints,” *IEEE Trans. Inf. Theory*, vol. 62, pp. 35–55, Jan 2016.

[36] 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.

[37] 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.

[38] S. Bubeck, *Convex Optimization: Algorithms and Complexity*, vol. 8. Foundations and Trends in Machine Learning, 2015.

[39] A. Rakhlin, O. Shamir, and K. Sridharan, “Making gradient descent optimal for strongly convex stochastic optimization,” *arXiv preprint arXiv:1109.5647*, 2011.

[40] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning - From Theory to Algorithms.* Cambridge University Press, 2014.

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 μ and ν defined on $\mathcal{Y}$, define $\phi(y) = (\mu(y) - e^{\varepsilon}\nu(y))_+$ and $\phi'(y) = (-\mu(y) + e^{\varepsilon}\nu(y))_+$ for any $y \in \mathcal{Y}$. Note that since $\frac{1}{2}\|\mu - e^{\varepsilon}\nu\|_1 = \mathsf{E}_{\varepsilon}(\mu\|\nu) + \frac{1}{2}(e^{\varepsilon} - 1)$ and $\|\phi\|_1 = \mathsf{E}_{\varepsilon}(\mu\|\nu)$, it follows that $\|\phi'\|_1 = \mathsf{E}_{\varepsilon}(\mu\|\nu) + e^{\varepsilon} - 1$. Letting E_{ε} denote $\mathsf{E}_{\varepsilon}(\mu\|\nu)$ for brevity, we can write

$$\begin{aligned}
\|\mu\mathsf{K} - e^{\varepsilon}\nu\mathsf{K}\|_1 &= \|(\mu(\mathrm{d}y) - e^{\varepsilon}\nu(\mathrm{d}y))\mathsf{K}\|_1 \\
&= \int_{\mathcal{Y}} \left| \int_{\mathcal{X}} (\mu(\mathrm{d}y) - e^{\varepsilon}\nu(\mathrm{d}y))\mathsf{K}(y) \right| \\
&= \int_{\mathcal{Y}} \left| \int_{\mathcal{X}} \phi(\mathrm{d}y)\mathsf{K}(y) - \int_{\mathcal{X}} \phi'(\mathrm{d}y')\mathsf{K}(y') \right| \\
&= \int_{\mathcal{Y}} \left| \|\phi\| \int_{\mathcal{X}} \frac{\phi(\mathrm{d}y)}{\|\phi\|}\mathsf{K}(y) - \|\phi'\| \int_{\mathcal{X}} \frac{\phi'(\mathrm{d}y')}{\|\phi'\|}\mathsf{K}(y') \right| \\
&= \int_{\mathcal{Y}} \Bigg| \|\phi\| \left(\int \frac{\phi'(\mathrm{d}y')}{\|\phi'\|} \right) \int_{\mathcal{X}} \frac{\phi(\mathrm{d}y)}{\|\phi\|}\mathsf{K}(y) \\
&\quad - \|\phi'\| \left(\int \frac{\phi(\mathrm{d}y)}{\|\phi\|}\mathrm{d}y \right) \int_{\mathcal{X}} \frac{\phi'(\mathrm{d}x')}{\|\phi'\|}\mathsf{K}(y') \Bigg| \\
&\le \max_{x,x'} \int_{\mathcal{Y}} \Big| \|\phi\|\mathsf{K}(y) - \|\phi'\|\mathsf{K}(y') \Big| \qquad (13) \\
&= \max_{x,x'} \int_{\mathcal{Y}} \Big| \mathsf{E}_{\varepsilon}\mathsf{K}(y) - (\mathsf{E}_{\varepsilon} + e^{\varepsilon} - 1)\mathsf{K}(y') \Big| \\
&\le \mathsf{E}_{\varepsilon} \max_{x,x'} \int_{\mathcal{Y}} \Big| \mathsf{K}(y) - e^{\varepsilon}\mathsf{K}(y') \Big| + (e^{\varepsilon} - 1)(1 - \mathsf{E}_{\varepsilon}). \qquad (14)
\end{aligned}$$

Notice that it follows from the definition of E_{ε}-divergence that

$$\mathsf{E}_{\varepsilon}(\mu\mathsf{K}\|\nu\mathsf{K}) = \frac{1}{2}\int |\mathrm{d}(\mu\mathsf{K} - e^{\varepsilon}\nu\mathsf{K})| - \frac{1}{2}(e^{\varepsilon} - 1).$$

Consequently, we obtain from (14) that

$$\mathsf{E}_{\varepsilon}(\mu\mathsf{K}\|\nu\mathsf{K}) \le \mathsf{E}_{\varepsilon}(\mu\|\nu) \max_{y,y'} \mathsf{E}_{\varepsilon}(\mathsf{K}(y)\|\mathsf{K}(y')),$$

and hence $\eta_{\varepsilon}(\mathsf{K}) \le \max_{y,y'} \mathsf{E}_{\varepsilon}(\mathsf{K}(y)\|\mathsf{K}(y'))$. 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 $\mu_{\delta} = \bar{\delta}\mathbb{I}_{\{y_0\}} + \delta\mathbb{I}_{\{y_1\}}$ and $\nu_{\delta} = (\bar{\delta}e^{-\varepsilon})\mathbb{I}_{\{y_0\}} + (1 - \bar{\delta}e^{-\varepsilon})\mathbb{I}_{\{y_2\}}$ where $\bar{\delta} := 1 - \delta$, $y_0 \notin \{y_1, y_2\}$ and $\mathbb{I}_{\{.\}}$ is the indicator function. It is easy to verify that $\mathsf{E}_{\varepsilon}(\mu_{\delta}\|\nu_{\delta}) = \delta$. We also have $\mu_{\delta}\mathsf{K} = \bar{\delta}\mathsf{K}(y_0) + \delta K(y_1)$ and $\nu_{\delta}\mathsf{K} = (\bar{\delta}/e^{\varepsilon})\mathsf{K}(y_0) + (1 - \bar{\delta}/e^{\varepsilon})\mathsf{K}(y_2)$. Hence, by (2),

$$\begin{aligned}
\mathsf{E}_{\varepsilon}(\mu_{\delta}\mathsf{K}\|\nu_{\delta}\mathsf{K}) &= \delta \int_{\mathcal{Y}} \left[\mathrm{d}(\mathsf{K}(y_1) - e^{\tilde{\varepsilon}}\mathsf{K}(y_2))(y) \right]_+ \\
&= \delta\mathsf{E}_{\tilde{\varepsilon}}(\mathsf{K}(y_1)\|\mathsf{K}(y_2)),
\end{aligned}$$

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

$$\eta_{\varepsilon}(\mathsf{K}) \ge \frac{\mathsf{E}_{\varepsilon}(\mu_{\delta}\mathsf{K}\|\nu_{\delta}\mathsf{K})}{\mathsf{E}_{\varepsilon}(\mu_{\delta}\|\nu_{\delta})} = \mathsf{E}_{\tilde{\varepsilon}}(\mathsf{K}(y_1)\|\mathsf{K}(y_2)).$$

By continuity of $\varepsilon \mapsto \mathsf{E}_{\varepsilon}(\mu\|\nu)$, we obtain from above

$$\eta_{\varepsilon}(\mathsf{K}) \ge \lim_{\delta \to 1} \mathsf{E}_{\tilde{\varepsilon}}(\mathsf{K}(y_1)\|\mathsf{K}(y_2)) = \mathsf{E}_{\varepsilon}(\mathsf{K}(y_1)\|\mathsf{K}(y_2)).$$

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

Proof of Theorem 2. Consider two neighboring datasets $x = \{x_1, \dots, x_n\}$ and $x' = \{x_1, \dots, x_{i-1}, x'_i, x_{i+1}, \dots, x_n\}$. Let μ_t and μ'_t be the distribution of W_t the output of the tth iteration when running on x and x', respectively. To derive δ for any given ε, we need to compute $\mathsf{E}_{\varepsilon}(\mu_T\|\mu'_T)$. Let $\pi : [n] \to [T]$ specifies an assignment of users to each batch, i.e., $\pi(i) = t$ if $i \in \mathsf{B}_t$. Note that $\mu_t = \mu'_t$ for $t < \pi(i)$. We now identify each iteration with a projected Markov kernel. At iteration t, the aggregator generates

$$W_t = \mathsf{proj}_{\rho}\Big(W_{t-1} - \frac{\eta_t}{m} \sum_{j \in \mathsf{B}_t} \nabla\ell(W_{t-1}, x_j) - \tilde{\sigma}_t Z_t \Big),$$

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

$$\Psi_t(w) = w - \frac{\eta_t}{m} \sum_{j \in \mathsf{B}_t} \nabla\ell(w, x_j).$$

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

$$\begin{aligned}
\mathsf{E}_{\varepsilon}(\mu_T\|\mu'_T) &\le \sum_{t=1}^{T} \Pr(\pi(i) = t)\mathsf{E}_{\varepsilon}(\mu_t\|\mu'_t) \prod_{j=t+1}^{T} \eta_{\varepsilon}(\mathsf{K}_j) \\
&= q\sum_{t=1}^{T} \mathsf{E}_{\varepsilon}(\mu_t\|\mu'_t) \prod_{j=t+1}^{T} \eta_{\varepsilon}(\mathsf{K}_j) \qquad (15)
\end{aligned}$$

To compute a bound for δ, it thus suffices to compute $\eta_{\varepsilon}(\mathsf{K}_j)$ for $j \in [T]$ and $\mathsf{E}_{\varepsilon}(\mu_t\|\mu'_t)$ for $t \in [T]$. We begin by computing $\eta_{\varepsilon}(\mathsf{K}_j)$ for $j \in [T]$ as follows

$$\begin{aligned}
\eta_{\varepsilon}(\mathsf{K}_j) &= \sup_{w_1, w_2 \in \mathsf{ball}(\rho)} \mathsf{E}_{\varepsilon}(\mathsf{K}_j(w_1)\|\mathsf{K}_j(w_2)) \\
&\le \sup_{w_1, w_2 \in \mathsf{ball}(\rho)} \mathsf{E}_{\varepsilon}(\mathcal{N}(\Psi_j(w_1), \tilde{\sigma}_j^2\mathrm{I})\|\mathcal{N}(\Psi_j(w_2), \tilde{\sigma}_j^2\mathrm{I})) \qquad (16) \\
&= \sup_{w_1, w_2 \in \Psi_j(\mathsf{ball}(\rho))} \mathsf{E}_{\varepsilon}(\mathcal{N}(w_1, \tilde{\sigma}_j^2\mathrm{I})\|\mathcal{N}(w_2, \tilde{\sigma}_j^2\mathrm{I})) \\
&= \theta_{\varepsilon}\Big(\frac{\Psi_j(\mathsf{ball}(\rho))}{\tilde{\sigma}_j} \Big) \qquad (17) \\
&\le \theta_{\varepsilon}\big(\frac{2\rho}{\tilde{\sigma}_j} \big) \qquad (18) \\
&= \theta_{\varepsilon}\Big(\frac{2\rho\sqrt{m}}{\eta_j\sigma_j} \Big) \qquad (19)
\end{aligned}$$

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

$$\begin{aligned}
&\mathsf{E}_{\varepsilon}(\mathsf{proj}_{\rho}(\mathcal{N}(\Psi_j(w_1), \tilde{\sigma}_j^2\mathrm{I}))\|\mathsf{proj}_{\rho}(\mathcal{N}(\Psi_j(w_2), \tilde{\sigma}_j^2\mathrm{I}))) \\
&\qquad \le \mathsf{E}_{\varepsilon}(\mathcal{N}(\Psi_j(w_1), \tilde{\sigma}_j^2\mathrm{I})\|\mathcal{N}(\Psi_j(w_2), \tilde{\sigma}_j^2\mathrm{I})),
\end{aligned}$$

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 β-smooth for all $x \in \mathcal{X}$, then $w \mapsto w - \nabla\ell(w, x)$ is contractive for $\eta \le \frac{2}{\beta}$ (see e.g., Prop 18 in [16]) and so is $w \mapsto \Psi_j(w)$; and (2) The map $r \mapsto \theta_{\varepsilon}(r)$ is increasing.

Next, we compute $\mathsf{E}_{\varepsilon}(\mu_t\|\mu'_t)$. Note that

$$\mu_t = \int_{\mathsf{ball}(\rho)} \mu_{t-1}(\mathrm{d}y)\mathsf{K}_t(y).$$

Since $\pi(i) = t$, data point $x'_i \in \mathsf{B}_t$. For this batch, we define

$$\Psi'_t(w) := w - \frac{\eta}{m}\Big[\nabla\ell(w, x'_i) + \sum_{j \in \mathsf{B}_t \setminus \{i\}} \nabla\ell(w, x_j) \Big],$$

and the corresponding Markov kernel K'_t associated with $w \mapsto \mathsf{proj}_{\rho}(\Psi'_t(w) - \tilde{\sigma}Z_t)$. It follows that

$$\mu'_t = \int_{\mathsf{ball}(\rho)} \mu_{t-1}(\mathrm{d}y)\mathsf{K}'_t(y).$$

The convexity of $(\mu, \nu) \mapsto \mathsf{E}_\varepsilon(\mu\|\nu)$ implies

$$
\begin{aligned}
\mathsf{E}_\varepsilon(\mu_t\|\mu_t') &\leq \int \mathsf{E}_\varepsilon(\mathsf{K}_t(y)\|\mathsf{K}_t'(y))\mu_{t-1}(\mathrm{d}y) && (20)\\
&\leq \int \mathsf{E}_\varepsilon(\mathcal{N}(\Psi_t(y), \tilde{\sigma}_t^2 \mathrm{I})\|\mathcal{N}(\Psi_t'(y), \tilde{\sigma}_t^2 \mathrm{I})\mu_{t-1}(\mathrm{d}y) && (21)\\
&= \int \theta_\varepsilon\Big(\frac{\|\Psi_t(y) - \Psi_t'(y)\|}{\tilde{\sigma}_t}\Big)\mu_{t-1}(\mathrm{d}y) && (22)\\
&\leq \theta_\varepsilon\Big(\frac{2L\eta_t}{m\tilde{\sigma}_t}\Big) && (23)\\
&\leq \theta_\varepsilon\Big(\frac{2L}{\sqrt{m}\sigma_t}\Big) && (24)
\end{aligned}
$$

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 $y \in \mathsf{ball}(\rho)$

$$
\begin{aligned}
\|\Psi_t(y) - \Psi_t'(y)\| &= \|\frac{\eta_t}{m}(\nabla\ell(y, x_i) - \nabla\ell(y, x_i'))\|\\
&\leq \frac{2L\eta_t}{m}
\end{aligned}
$$

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

$$
\mathsf{E}_\varepsilon(\mu_T\|\mu_T') \leq q\sum_{t=1}^{T}\theta_\varepsilon\left(\frac{2L}{\sqrt{m}\sigma_t}\right)\prod_{j=t+1}^{T}\theta_\varepsilon\left(\frac{2\rho\sqrt{m}}{\eta_j\sigma_j}\right).
$$

Assuming $\eta_t = \eta$ and $\sigma_t = \sigma$ for $t \in [T]$, the above upper-bound can be simplified as

$$
\begin{aligned}
\mathsf{E}_\varepsilon(\mu_T\|\mu_T') &\leq q\theta_\varepsilon\Big(\frac{2L\sqrt{m}}{\sigma}\Big)\sum_{t=1}^{T}\theta_\varepsilon^{T-t}\Big(\frac{2\rho\sqrt{m}}{\eta\sigma}\Big)\\
&= q\theta_\varepsilon\Big(\frac{2L\sqrt{m}}{\sigma}\Big)\frac{1 - \theta_\varepsilon\left(\frac{2\rho\sqrt{m}}{\eta\sigma}\right)^{\frac{n}{m}}}{1 - \theta_\varepsilon\left(\frac{2\rho\sqrt{m}}{\eta\sigma}\right)}.
\end{aligned}
$$

The desired result then follows by invoking (3). ∎

Proof of Corollary 2. Observe that

$$
\begin{aligned}
&\mathbb{E}\left[\|W_{t+1}\|_2^2\right]\\
=&\mathbb{E}\left[\left\|\mathsf{Proj}_\rho\left(W_t - \eta_t\left(\nabla f_{\sigma(t)}\left(W_t\right) + \sigma_t Z_t\right)\right)\right\|_2^2\right]\\
\leq&\mathbb{E}\left[\left\|W_t - \eta_t\left(\nabla f_{\sigma(t)}\left(W_t\right) + \sigma_t Z_t\right)\right\|_2^2\right]\\
\leq&\mathbb{E}\left[\left\|W_t - \eta_t\nabla f_{\sigma(t)}\left(W_t\right)\right\|_2^2\right] + \eta_t^2\mathbb{E}\left[\|\sigma_t Z_t\|_2^2\right]\\
\leq&\mathbb{E}\left[\|W_t\|^2\right] - 2\eta_t\mathbb{E}\left[\langle\nabla f_{\sigma(t)}\left(W_t\right), W_t\rangle\right] + \eta_t^2\left(G^2 + \sigma_t^2\right).
\end{aligned}
$$

Follow the rest of the proof in Theorem 3 of [20], we obtain the desired result. ∎

Proof of Corollary 3. Following the analysis in Theorem 2, let μ_τ and μ_τ' be the distributions of W_τ with two respect to two neighboring datasets x and x' which differ at the i-th item. Then we have

$$
\begin{aligned}
&\mathsf{E}_\varepsilon(\mu_\tau\|\mu_\tau')\\
&= \mathsf{E}_\varepsilon\left(\frac{1}{T}\sum_{\tau=1}^{T}\mu_\tau\|\frac{1}{T}\sum_{\tau=1}^{T}\mu_\tau'\right) \leq \frac{1}{T}\sum_{\tau=1}^{T}\mathsf{E}_\varepsilon(\mu_\tau\|\mu_\tau')\\
&\overset{\text{(a)}}{\leq} \frac{1}{T}\sum_{\tau=1}^{T}\sum_{t=1}^{T}\mathbb{1}_{\{t\leq\tau\}}\Pr(\pi(i) = t)\mathsf{E}_\varepsilon(\mu_t\|\mu_t')\prod_{j=t+1}^{\tau}\eta_\varepsilon(\mathsf{K}_j)\\
&= \frac{1}{T^2}\sum_{\tau=1}^{T}\sum_{t=1}^{\tau}\mathsf{E}_\varepsilon(\mu_t\|\mu_t')\prod_{j=t+1}^{\tau}\eta_\varepsilon(\mathsf{K}_j)\\
&\overset{\text{(b)}}{\leq} \frac{1}{T^2}\sum_{\tau=1}^{T}\sum_{t=1}^{\tau}\theta_\varepsilon\left(\frac{2L}{\sqrt{m}\sigma_t}\right)\prod_{j=t+1}^{\tau}\theta_\varepsilon\left(\frac{2\rho\sqrt{m}}{\eta_j\sigma_j}\right)
\end{aligned}
$$

where (a) follows from Jensen's inequality and the convexity of E_ε, and (b) is from (19) and (24). If we set $\eta^* \triangleq \min_{t\in[T]}\eta_t$ and $\sigma^* \triangleq \min_{t\in[T]}\sigma_t$, then $\mathsf{E}_\varepsilon(\mu_\tau\|\mu_\tau')$ can be further bounded by

$$
\begin{aligned}
&\frac{1}{T^2}\sum_{\tau=1}^{T}\sum_{t=1}^{\tau}\theta_\varepsilon\left(\frac{2L}{\sqrt{m}\sigma_t}\right)\prod_{j=t+1}^{\tau}\theta_\varepsilon\left(\frac{2\rho\sqrt{m}}{\eta_j\sigma_j}\right)\\
&\leq \frac{1}{T^2}\theta_\varepsilon\left(\frac{2L}{\sqrt{m}\sigma^*}\right)\sum_{\tau=1}^{T}\sum_{t=1}^{\tau}\theta_\varepsilon^{\tau-t}\left(\frac{2\rho\sqrt{m}}{\eta^*\sigma^*}\right)\\
&= \frac{1}{T^2}\theta_\varepsilon\left(\frac{2L}{\sqrt{m}\sigma^*}\right)\sum_{\tau=1}^{T}\frac{1 - \theta_\varepsilon^\tau\left(\frac{2\rho\sqrt{m}}{\eta^*\sigma^*}\right)}{1 - \theta_\varepsilon\left(\frac{2\rho\sqrt{m}}{\eta^*\sigma^*}\right)}\\
&= \frac{1}{T^2}\theta_\varepsilon\left(\frac{2L}{\sqrt{m}\sigma^*}\right)\sum_{\tau=1}^{T}\frac{1 - \theta_\varepsilon^\tau\left(\frac{2\rho\sqrt{m}}{\eta^*\sigma^*}\right)}{1 - \theta_\varepsilon\left(\frac{2\rho\sqrt{m}}{\eta^*\sigma^*}\right)}.
\end{aligned}
$$

∎