Private Weighted Random Walk Stochastic Gradient Descent

Ghadir Ayache and Salim El Rouayheb

***Abstract*—We consider a decentralized learning setting in which data is distributed over nodes in a graph. The goal is to learn a global model on the distributed data without involving any central entity that needs to be trusted. While gossip-based stochastic gradient descent (SGD) can be used to achieve this learning objective, it incurs high communication and computation costs. To speed up the convergence, we propose instead to study random walk based SGD in which a global model is updated based on a random walk on the graph. We propose two algorithms based on two types of random walks that achieve, in a decentralized way, uniform sampling and importance sampling of the data. We provide a non-asymptotic analysis on the rate of convergence, taking into account the constants related to the data and the graph. Our numerical results show that the weighted random walk based algorithm has a better performance for high-variance data. Moreover, we propose a privacy-preserving random walk algorithm that achieves local differential privacy based on a Gamma noise mechanism that we propose. We also give numerical results on the convergence of this algorithm and show that it outperforms additive Laplace-based privacy mechanisms.**

***Index Terms*—Decentralized learning, random walk, importance sampling, local differential privacy.**

I. Introduction

WE CONSIDER the problem of designing a decentralized learning algorithm on data that is distributed among the nodes of a graph. Each node in the graph has some local data, and we want to learn a model by minimizing an empirical loss function on the collective data. We focus on applications such as decentralized federated learning or Internet-of-Things (IoT) networks, where the nodes, being phones or IoT devices, have limited communication and energy resources. A crucial constraint that we impose is excluding the reliance on any central unit (parameter server, aggregator, etc.) that can communicate with all the nodes and orchestrate the learning algorithm. Therefore, we seek decentralized algorithms that are based only on local communication between neighboring nodes. Privacy is our main motivation for such algorithms since nodes do not have to trust a central unit. Naturally, decentralized algorithms come with the added benefits of avoiding single points of failure and easily adapting to dynamic graphs.

Gossip-style gradient descent algorithms, e.g., [1], are a major contender for solving our problem. In Gossip algorithms, each node has a *local* model that is updated and exchanged with the neighboring nodes in each iteration. Gossip algorithms can be efficient time-wise since nodes can update their model in parallel. However, it incur high communication and energy (battery) cost in order to wait for all the local models to converge. This diminishes the appeal of Gossip algorithms for the applications we have in mind. Instead, we propose to study decentralized algorithms based on random walks. A random walk passes around a global model, and in each iteration, activates a node which updates the global model based on its local data. The activated node then passes the updated global model to a randomly chosen neighbor. The random walk guarantees that every cost spent on the communication and computation resources goes to improve the global model. Therefore, it outperforms Gossip algorithms in terms of these costs. In this article, we will study how the design of the random walk affects the convergence of the learning algorithm. Namely, we propose two algorithms that we call Uniform Random Walk SGD and Weighted Random Walk SGD and study their convergence. Moreover, we consider the setting in which nodes do not completely trust their neighbors and devise locally differential variants of these algorithms.

A. Previous Work

The two works that are most related to our work are [2] and [3]. The work in [2] studies the random walk data sampling for stochastic gradient descent (SGD). The work of [3] focuses on importance sampling to speed up the convergence; however, it is implemented for a centralized setting.

A line of work on random walk based decentralized algorithm has been studied in the literature under the name of incremental methods. Early works on incremental methods by [4]–[6] have established theoretical convergence guarantees for different convex problem settings and using first-order methods of stochastic gradient updates. In [7], more advanced stochastic updates is been employed, namely CIAG [8], to improve the convergence guarantees for strongly convex objective by implementing unbiased total gradient tracking technique that uses Hessian information to accelerate convergence; they show that their algorithm converges linearly with high probability. The work of [2] proposed to speed-up the convergence by using non-reversible random walks.

Manuscript received August 15, 2020; revised December 4, 2020; accepted January 14, 2021. Date of publication January 22, 2021; date of current version March 16, 2021. *(Corresponding author: Ghadir Ayache.)*

Ghadir Ayache is with the Department of Electrical and Computer Engineering, Rutgers University, Piscataway, NJ 08854 USA (e-mail: ghadir.ayache@rutgers.edu).

Salim El Rouayheb is with the Department of Electrical and Computer Engineering, Rutgers University, New Brunswick, NJ 08901 USA (e-mail: salim.elrouayheb@rutgers.edu).

Digital Object Identifier 10.1109/JSAIT.2021.3052975

In [9], the authors studied the convergence analysis of random walk learning algorithm for the alternating direction method of multipliers (ADMM).

Gossip-based algorithms are also popular approach that has been widely studied, with application to averaging [10] and various stochastic gradient learning [6], [11]–[13].

In terms of privacy, many works study differential privacy mechanisms for SGD algorithms. One approach for privacy is local differential privacy (LDP) [14], [15], where privacy is applied locally, which is well-suited for decentralized algorithms. Privacy mechanisms for SGD suggest protecting the output of every iteration by output perturbation [16] or by gradient perturbation and quantization (e.g., [17]–[20]).

B. Contributions

In our work, we are interested in decentralized settings where the nodes have limited computation and communication power. Moreover, we assume that the nodes do not fully trust each others with their data and require privacy guarantees. For these reasons, we focus on first-order methods where the information exchanged between neighboring nodes is restricted to the model at a given iteration, serving both our communication and privacy constraints. Moreover, the update step involves only the gradient of the local loss function but no higher order derivatives, limiting the computation cost at the nodes. In this constrained setting, we study the design the random walk to speed up the convergence by simulating importance sampling.

It is known that the natural random walk on the vertices of a graph that picks one of the neighbors uniformly at random, will end up visiting nodes proportionally to their degree. The consequence of this is that the underlying SGD learning algorithm will be effectively sampling with higher probability the data on highly connected nodes. This sampling, which is biased by the graph topology, may or may not be a good choice for speeding-up the convergence of the algorithm, depending on which nodes have the important data.

We propose two alternate Markov-Hasting sampling schemes to address this bias, and study the convergence of the resulting algorithms: (i) *Uniform Random Walk SGD* Algorithm 1, in which all the nodes are visited equally likely in the stationary regime irrespective of the node degrees. This emulates uniform data sampling in the centralized case; (ii) *The Weighted Random Walk SGD* Algorithm 2, in which the nodes are sampled proportional to the gradient-Lipschitz constant of their local loss function, rather than their degree. This emulates importance sampling [3], [21] in the centralized case.

We study the rate of convergence of both algorithms. The asymptotic rate of convergence of both algorithms is same as that of the natural random walk, namely, $\mathcal{O}(\frac{1}{k^{1-q}})$, where k is the number of iterations and $q \in (\frac{1}{2}, 1)$. However, the constants are different and depend on the gradient-Lipschitz constants of the local loss functions and the graph spectral properties. Our numerical simulations highlight how these constants affect the non-asymptotic behavior of these algorithms. we observe that in high variance data regime Algorithm 2 outperforms Algorithm 1, and the opposite happens in a low variance data regime. These observations are in conformity with the behavior of importance sampling in [3] for the centralized case. This is expected since the random walks were designed to achieve, in the stationary regime, the same sampling strategy of the centralized case. The challenge, however, is with the proof techniques, which build on the random walk learning results in [2] and importance sampling in [3].

In addition, we propose a third algorithm, Private Weighted Random Walk SGD (Algorithm 3), that ensures local differential privacy of the local data. We propose a privacy mechanism that uses Gamma noise and characterize the tradeoff between the noise level and the privacy level. We also give numerical results on the convergence of Algorithm 3 and show that it outperforms additive Laplace-based privacy mechanisms.

Parts of these results pertaining to Algorithms 1 and 2 have already appeared in a conference paper [22].

C. Organization

The rest of this article is organized as follows. We formulate the problem in Section II. We present Algorithms 1 and 2 in Section III. We present the convergence result of both algorithms in Section III-C. In Section IV, we define the Gamma mechanism and propose a privacy-preserving version of Algorithm 2. Finally, we present numerical results of our algorithms in Section V. We conclude in Section IV. The proofs of the technical results are deferred to the appendices.

II. Problem Description

A. Problem Setting

Network Model: We consider a communication network represented by an undirected graph $\mathcal{G}(V, E)$ with $V = [N] := \{1, \ldots, N\}$ is the set of nodes and $E \subseteq V \times V$ is the set of edges. Two nodes in $\mathcal{G}$ connected by an edge in E can communicate and exchange information. We refer to such two nodes as neighbors. We denote the set of neighbors of a node $i \in V$ by $\mathcal{N}(i)$, and its degree by $\deg(i) := |\mathcal{N}(i)|$. We assume that the graph $\mathcal{G}$ is connected, i.e., there exits a path between any two nodes in V. Moreover, we assume the presence of a self-loop at every node, i.e., $\{(i, i) : i = 1, 2, \ldots, N\} \subset E$. We assume that the set $\mathcal{N}(i)$ does not contain i.

Data and Learning Objective: Each node $i \in [N]$ has a feature vector $x_i \in \mathcal{X} \subseteq \mathbb{R}^d$, where $\mathcal{X}$ is the feature space and d is the number of features. Moreover, node i has a label $y_i \in \mathbb{R}$ corresponding to x_i.

The objective is to learn a prediction function on the collective data distributed over all the nodes by learning a global model $w \in \mathcal{W}$, where $\mathcal{W}$ is a closed and bounded subset in $\mathbb{R}^d$. Thus, an optimal model w^* solves

$$\underset{w \in \mathcal{W}}{\text{minimize}} \frac{1}{N} \sum_{i=1}^{N} f_i(w; x_i, y_i), \tag{1}$$

where $f_i(.)$ is the local loss function at node i. We denote the global loss function on all the data by

$$f(w) = \frac{1}{N} \sum_{i=1}^{N} f_i(w; x_i, y_i).$$

We will assume convexity and gradient-Lipschitzness of the local loss functions as described in Assumption 1.

Assumption 1: The local loss function f_i for each node $i \in V$ is a convex function on $\mathcal{W} \in \mathbb{R}^d$ and has an L_i-Lipschitz continuous gradient; that is, for any $w, w' \in \mathcal{W}$, there exists a constant $L_i > 0$ such that

$$\left\|\nabla f_i(w) - \nabla f_i\left(w'\right)\right\|_2 \leq L_i \left\|w - w'\right\|_2.$$

To solve the optimization problem in (1), we perform an iterative stochastic gradient descent (SGD) method. At each iteration k, one random node i in the network will perform a gradient descent update and project the result back into the feasible set $\mathcal{W}$ as follows:

$$w^{(k+1)} = \mathbf{\Pi}_{\mathcal{W}}\left(w^{(k)} - \gamma^{(k)} \hat{\nabla} f\left(w^{(k)}\right)\right), \tag{2}$$

where $\gamma^{(k)}$ is the step size, $w^{(k)}$ is the global model update at iteration k, and $\hat{\nabla} f(w^{(k)})$ is a gradient estimate of the global loss function $f(.)$ based on one node i local data. For convergence guarantees [23], we use a decreasing step size that satisfies

$$\sum_{k=1}^{\infty} \gamma^{(k)} = +\infty, \quad \text{and} \quad \sum_{k=1}^{\infty} \ln k . \left(\gamma^{(k)}\right)^2 < +\infty. \tag{3}$$

In particular, we will take $\gamma^{(k)} = \frac{1}{k^q}$ for $0.5 < q < 1$ which satisfy the conditions in (3).

Our goal is to implement a decentralized algorithm based on a Markov random walk sampling decision on the graph $\mathcal{G}$ to solve the learning problem in (1) with privacy guarantees, as explained in the next two sections.

B. Random Walk Learning Algorithm

We seek a decentralized algorithm that will learn the model w^* without involving a central entity, and in which nodes exchange information only with their neighbors. Towards that goal, we design a random walk on $\mathcal{G}$ that activates the node that it visits to update the global model based on the local data of the activated node. The algorithm is initiated at one node picked uniformly at random to be activated at the first iteration to perform the update in (2). Afterwards, at every iteration $k = 1, 2, \ldots$, the global model iterate $w^{(k)}$ is passed to a randomly chosen node $i^{(k)} \in V$ to update it using its local data. Given the connection constraint, at iteration k, the activated node $i^{(k)}$ is a neighbor node of the previously activated node $i^{(k-1)}$, i.e., $i^{(k)} \in \mathcal{N}(i^{(k-1)})$. The process above defines a random walk on $\mathcal{G}$ that we model by a Markov chain. The sequence of active nodes $i^{(k)}$ are the states of the Markov chain with state-space V and transition matrix P that inherits the same connection structure of the underlying connected graph $\mathcal{G}$. For convergence guarantees, we will make the next assumption:

Assumption 2: The Markov chain $(i^{(k)})_{k\in\mathbb{N}}$ defined on the finite state space V with homogeneous transition matrix P is irreducible and aperiodic and has a stationary distribution π on V.

The problem is to design the transition probabilities in P to speed up the convergence of the decentralized algorithm by only using the local information available to the nodes, namely their degrees and Lipschitz constants.

C. Local Differential Privacy

In addition to designing a decentralized random walk learning algorithm that speeds up convergence, we aim for a privacy-preserving algorithm to protect each node data from being revealed by any other node including its neighbors.

We adopt local differential privacy (LDP) [14], [15] as a privacy measure that is well-suited to our decentralized setting, since we excluded the involvement of any central aggregator that would coordinate the learning process. Accordingly, nodes will share a noisy version of the messages that needs to be exchanged with the neighbors in a way that ensures a desired level of privacy for the nodes' local data.

Consider a message $M(x_i)$ that is to be sent from node i to one of its neighbors. M is a function of the node's local data $M : \mathcal{X} \to \mathbb{M}$, where $\mathbb{M}$ is the image space. Let $\mathcal{R} : \mathbb{M} \to Img\mathcal{R}$ be the privatization scheme that node i will perform on the message to share. For any two possible data points $x_i, x_i' \in \mathcal{X}$ that could be owned by a node i, the (ϵ, δ)-LDP defined as follows:

Definition 1 (Local Differential Privacy) [14]: A randomization mechanism $\mathcal{R}$ is (ϵ, δ)-LDP, if for any $x_i, x_i' \in \mathcal{X}$ and for any $r \in Img\mathcal{R}$, we have

$$P(\mathcal{R}(M(x_i)) = r) \leq e^{\epsilon} P\left(\mathcal{R}\left(M\left(x_i'\right)\right) = r\right) + \delta, \tag{4}$$

where $\epsilon \geq 0$ quantifies the privacy level, and $\delta \in [0, 1]$ quantifies the allowed probability of violating the privacy bound [24], [25].

A drawback of LDP noising mechanisms is that it will affect the utility of the exchanged messages and, as a result, slow the convergence of decentralized learning algorithm. We propose an LDP mechanism based on Gamma noise that achieves an attractive tradeoff between privacy and convergence, and outperforms generic additive-noise (Gaussian or Laplace) LDP mechanisms.

For ease of reference, we summarize our notation in Table I.

III. Decentralized Weighted Random Walk SGD

We focus in this section on the design of the random walk learning algorithm without the privacy constraint. To decide on the transition probabilities of the random walk, a natural choice would be to pick the next node uniformly at random from the neighbors of the current active node. This gives a stationary distribution π proportional to the degree of each node, i.e., $\pi(i) \sim \deg(i)$. Therefore, Assumption 2 implies the fraction of time a node is activated is directly proportional to its degree [26]. The effect of this random walk on the learning algorithm is that, in the update step of (2) the data is sampled proportional to the degree of the node that is carrying it. This node degree biased sampling may not be a favorable choice for speeding up the convergence of our decentralized learning algorithm. To counterbalance this bias, we propose two alternate sampling schemes and study the convergence of the resulting algorithms.

TABLE I
NOTATION

$\mathcal{G}$	Graph representing the network
N	Total number of nodes in $\mathcal{G}$
V	Set of nodes in $\mathcal{G}$
E	Set of edges in $\mathcal{G}$
$\deg(i)$	Degree of node i
x_i	Data feature vector of node i
y_i	Data label of node i
f_i	Local loss function of node i
L_i	Gradient-Lipschitz constant of f_i
f	Global loss function
w	Data model
w^*	Data optimal model
$w^{(k)}$	Model update at iteration k
$\sup L$	Maximum gradient-Lipschitz constant
$\inf L$	Minimum gradient-Lipschitz constant
$\bar{L}$	Average gradient-Lipschitz constant: $\bar{L} = \frac{L_1+L_2+\ldots+L_N}{N}$
$i^{(k)}$	Node visited at time k
ϵ	Privacy level
$1-\delta$	Privacy confidence level
π_u	Stationary distribution for uniform random walk
P_u	Transition matrix for uniform random walk
π_w	Stationary distribution for weighted random walk
P_w	Transition matrix for weighted random walk
$\mathcal{R}(L_i)$	Noisy version of L_i, output of the privacy mechanism
$\pi_{w,\mathcal{R}}$	Stationary distribution for noisy (private) weighted random walk

1) *Uniform Random Walk SGD*, in which all the nodes are visited equally likely in the stationary regime irrespective of their degree. This guarantees that the data points are eventually sampled uniformly at random irrespective of the graph topology.
2) *Weighted Random Walk SGD*, in which the nodes are sampled proportional to the gradient-Lipschitz constant of their local loss function, rather than their degree. The motivation is to achieve importance sampling [3], [21] in a decentralized fashion.

A. Uniform Random Walk SGD

The Uniform Random Walk SGD algorithm aims at sampling the data points held by the graph nodes uniformly at random, similarly to what is done in standard centralized SGD [27]–[29]. We achieve the uniform data sampling by designing a random walk with a transition matrix P_u using only local node information, such that the chain converges to the stationary distribution π_u such that

$$\pi_u(i) = \frac{1}{N}, \quad \forall i \in V.$$

We implement the Metropolis Hasting (MH) decision rule to design the transition probabilities, so the random walk converges to the desired stationary π_u [2], [26], [30]. The MH rule can be described as follows:

1) At the k^{th} step of the random walk, the active node $i^{(k)}$ selects uniformly at random one of its neighbors, say j, as a candidate to be the next active node. This selection gets accepted with probability

$$a_u\left(i^{(k)}, j\right) = \min\left(1, \frac{\deg\left(i^{(k)}\right)}{\deg(j)}\right).$$

Upon the acceptance, we have $i^{(k+1)} = j$.

Algorithm 1 Uniform Random Walk SGD

Initialization: Initial node $i^{(1)}$ chosen uniformly at random from $[N]$, Initial model $w^{(1)}$ chosen uniformly at random from $\mathcal{W}$
for $k = 1$ **to** T **do**
  $\hat{\nabla} f\left(w^{(k)}\right) = \nabla f_i\left(w^{(k)}\right)$.
  $w^{(k+1)} = \Pi_{\mathcal{W}}\left(w^{(k)} - \gamma^{(k)} \hat{\nabla} f\left(w^{(k)}\right)\right)$
  Choose node j uniformly at random from $\mathcal{N}\left(i^{(k)}\right)$.
  Generate $p \sim U(0, 1)$ where U is the uniform distribution.
  if $p \leq \min\left\{1, \frac{\deg\left(i^{(k)}\right)}{\deg(j)}\right\}$ **then**
    $i^{(k+1)} \leftarrow j$
  else
    $i^{(k+1)} \leftarrow i^{(k)}$
  end if
end for
Return: $w^{(T)}$.

2) Otherwise, if the candidate node gets rejected, the random walk stays at the same node, i.e., $i^{(k+1)} = i^{(k)}$.

Therefore, the transition matrix P_u is given by

$$P_u(i, j) = \begin{cases} \frac{1}{\deg(i)} \min\left\{1, \frac{\deg(i)}{\deg(j)}\right\} & j \neq i \text{ and } j \in \mathcal{N}(i), \\ 1 - \sum_{j \in \mathcal{N}(i)} \frac{1}{\deg(i)} \min\left\{1, \frac{\deg(i)}{\deg(j)}\right\} & j = i, \text{ and} \\ 0 & \text{otherwise.} \end{cases}$$

This random walk does conform with Assumption 2 and it uses local information only. By the Ergodic theorem [26] the above random walk that converges to a uniform stationary distribution, samples all the states (nodes) uniformly at random on the long-term run.

The Uniform Random Walk SGD implements the uniform random walk through the MH rule above. And, once a node is activated it updates the global model based on (2).

In Section III-C, we give the convergence analysis for this algorithm.

B. Weighted Random Walk SGD

To speed up the convergence, we propose a decentralized sampling that mimics centralized importance sampling, which consists of selecting more often the more informative data points [3], [21], [31]–[34]. In our analysis, we will take the data importance to be reflected through the gradient-Lipschitz constant of the node's local loss function [3]. Again, we utilize the MH decision rule for the random walk to achieve a stationary distribution proportional to the node gradient-Lipschitz constant as follows

$$\pi_w(i) = \frac{L_i}{\sum_{j=1}^{N} L_j} \quad \forall i \in V.$$

The random walk proceeds as previously explained, except for the probability of accepting the candidate node, which is now

$$a_w\left(i^{(k)}, j\right) = \min\left(1, \frac{L_j}{L_{i^{(k)}}} \frac{\deg\left(i^{(k)}\right)}{\deg(j)}\right). \tag{5}$$

Algorithm 2 Weighted Random Walk SGD

Initialization: Initial node $i^{(1)}$ chosen uniformly at random from $[N]$, Initial model $w^{(1)}$ chosen uniformly at random from $\mathcal{W}$

Every node i shares L_i and $\deg(i)$ it with its neighbors.

for $k = 1$ **to** T **do**

$\hat{\nabla} f\big(w^{(k)}\big) = \frac{\bar{L}}{L_{i^{(k)}}} \nabla f_{i^{(k)}}\big(w^{(k)}\big)$

$w^{(k+1)} = \Pi_{\mathcal{W}}\Big(w^{(k)} - \gamma^{(k)} \hat{\nabla} f\big(w^{(k)}\big)\Big)$

Choose node j uniformly at random from $\mathcal{N}\big(i^{(k)}\big)$.

Generate $p \sim U(0, 1)$ where U is the uniform distribution.

if $p \leq \min\left\{1, \frac{L_j}{L_{i^{(k)}}} \frac{\deg\big(i^{(k)}\big)}{\deg(j)}\right\}$ **then**

$i^{(k+1)} \leftarrow j$

else

$i^{(k+1)} \leftarrow i^{(k)}$

end if

end for

Return: $w^{(T)}$.

Consequently, the new transition matrix P_w generating the weighted random walk is given by

$$P_w(i,j) = \begin{cases} \frac{1}{\deg(i)} \min\left\{1, \frac{L_j}{L_i} \frac{\deg(i)}{\deg(j)}\right\} & j \neq i \text{ and } j \in \mathcal{N}(i), \\ 1 - \sum_{j \in \mathcal{N}(i)} \frac{1}{\deg(i)} \min\left\{1, \frac{L_j}{L_i} \frac{\deg(i)}{\deg(j)}\right\} & j = i, \text{ and} \\ 0 & \text{otherwise.} \end{cases}$$

C. Convergence Analysis

We summarize here our main results on the convergence of Algorithms 1 and 2. Our aim is to characterize how the design of the random walk affects the non-asymptotic rate of convergence of the algorithms. To give a bit of context, for convex and gradient-Lispchitz objective functions, centralized SGD has an asymptotic convergence rate in the order of $\mathcal{O}(\frac{1}{\sqrt{k}})$ (after k iterations) for diminishing step-size and for independent data sampling [35], [36]. For our decentralized setting, in which data sampling is not independent and is constrained by the graph topology, random walk SGD algorithms can approach the same rate of convergence for convex global loss function [1], [2], [4], [5], [37]. Namely, for a step-size $\gamma^{(k)} = \frac{1}{k^q}$ with $\frac{1}{2}, < q < 1$, the work in [2] proves a rate of convergence $\mathcal{O}(\frac{1}{k^{1-q}})$.

It was shown in [37] that $\Omega(\frac{1}{\sqrt{k}})$, k being the number of SGD updates, is a fundamental lower bound on the convergence rate for convex optimization within the stochastic first-order oracle model with no access to independent sampling.

Our proposed algorithms will also have this rate of convergence. However, the choice of the random walk will affect the constants in the rate of convergence and can lead to a speed-up in the non-asymptotic regime. Theorems 1 and 2 characterize these constants of our proposed algorithms.

We denote the eigenvalues of any the transition matrix P of the random walk by $\lambda_1 = 1 > \lambda_2 \geq \cdots \geq \lambda_N$, and define $\lambda_P = \frac{\max\{|\lambda_2|, |\lambda_N|\}+1}{2}$.

Theorem 1 (Convergence of Algorithm 1): Under Assumptions 1 and 2, the Uniform Random Walk SGD (Algorithm 1) converges in the mean sense, i.e.,

$$\lim_{k\to\infty} \mathbb{E}\Big(f\Big(w^{(k)}\Big) - f\big(w^*\big)\Big) = 0. \tag{6}$$

Moreover, its rate of convergence, for a step size $\gamma^{(k)} = \frac{1}{k^q}$, $0.5 < q < 1$, is

$$\mathbb{E}\Big[f\Big(\bar{w}^{(k)}\Big) - f\big(w^*\big)\Big] = O\left(\frac{\max\Big\{\sigma^2, (\sup L)^2, \frac{1}{\ln(1/\lambda_{P_u})}\Big\}}{k^{1-q}}\right), \tag{7}$$

where, $\bar{w}^{(k)} = \frac{1}{\sum_{n=1}^{k} \gamma^{(n)}} \sum_{m=1}^{k} \gamma^{(m)} w^{(m)}$, $\sup L = \max_{i\in V} L_i$ and the residual $\sigma^2 = \max_{w^*} \mathbb{E}[\|\nabla f_i(w^*)\|_2^2]$.

Note the technicality that the convergence is in terms of $\bar{w}^{(k)}$, instead of $w^{(k)}$, which is weighted average of the $w^{(k)}$s. This is due to the proof technique. Next, we give the bound on the convergence rate for the Weighted Random Walk SGD.

Theorem 2 (Convergence Rate of Algorithm 2): Under assumptions 1 and 2, the Weighted Random Walk SGD (Algorithm 2) converges in the mean sense, i.e.,

$$\lim_{k\to\infty} \mathbb{E}\Big(f\Big(w^{(k)}\Big) - f\big(w^*\big)\Big) = 0. \tag{8}$$

Moreover, its rate of convergence, for a step size $\gamma^{(k)} = \frac{1}{k^q}$ when $0.5 < q < 1$, is

$$\mathbb{E}\Big[f\Big(\bar{w}^{(k)}\Big) - f\big(w^*\big)\Big] = O\left(\frac{\max\Big\{\frac{\bar{L}}{\inf L}\sigma^2, \big(\bar{L}\big)^2, \frac{1}{\ln(1/\lambda_{P_w})}\Big\}}{k^{1-q}}\right), \tag{9}$$

where $\inf L = \min_{i\in V} L_i$ and $\bar{L} = \frac{\sum_i L_i}{N}$.

The results on the rate of convergence stated in the above theorems show the tradeoff that the two random walks offer. The weighted sampling for random walk SGD improves the bound on convergence from being a function of $\sup L$ to be a function of the average $\bar{L}$. However, it amplifies the effect of the residual represented by σ^2. The numerical results will show later that in high variance data regime resulting in a wide range for the values of the Lipschitz constants, the improvement brought by $\bar{L}$ dominates over the residual amplification, so Algorithm 2 outperforms Algorithm 1. However, the opposite happens for a low variance data regime where the data variance is low, and therefore the gap between $\sup L$ and $\bar{L}$ is not significant. Therefore, these observations are in conformity with the behavior of importance sampling in [3] for the centralized case. The proof of Theorem 1 follows the same steps as in [2] with incorporation of the Lipschitzness assumption and we provide it for completeness and for comparison with Theorem 2. The detailed proofs for both theorems can be found in Appendices A and B.

D. Comparison to Gossip Algorithm

To better understand the performance of our two algorithms, we compare them to Gossip algorithm. One of the most common approaches of the Gossip algorithms is the synchronous version [6], [31], [38]–[40], in which at every round, all nodes exchange and update their local models. For convex problems, the convergence rate is $O(1/\sqrt{k})$ [41], where k is the algorithm round in which $|\mathcal{E}|$ communication links are activated and $O(N)$ computations are performed. Therefore, the convergence rate as function of communication cost is $O(|\mathcal{E}|/\sqrt{k})$, and as a function of computation cost is $O(N/\sqrt{k})$. However, the convergence rate of Algorithms 1 and 2 as a function of computation and communication cost approaches a rate of $O(1/\sqrt{k})$. That is because at each round at most[1] one communication link is activated and one SGD update is performed.

IV. Differential Private Random Walk SGD

In this section, we propose an LDP mechanism to be applied to the messages exchanged within Algorithm 2 to make it privacy-preserving. Algorithm 2 requires sharing two pieces of information between two neighbouring nodes:

1) The gradient-Lipschitz constants: Algorithm 2 requires every node $i \in V$ to share the gradient-Lipschitz constant L_i with all its neighbors to be able to implement the MH rule of (5). This can leak important information about the local data at the nodes. For instance, for linear regression loss function, we have $L_i = N\|x_i\|_2^2$.
2) The model updates: An activated node needs to receive the latest updated version of the model $w^{(k)}$, which naturally contains information about the data on the nodes visited so far.

The literature on privacy-preserving SGD has extensively studied the problem of designing LDP mechanisms for the model update or the gradient through model perturbation (e.g., [16]) or gradient perturbation and quantization (e.g., [17]–[20]). Within that space of works, the new aspect of our problem is the sharing of the gradient-Lipschitz constants. For this reason, we focus in our analysis on characterizing how sharing privatized (noisy) versions of these constants would affect the random walk learning algorithm, and assume that the models are shared in the clear.[2]

We propose Algorithm 3 which is a modification of Algorithm 2 that makes it privacy-preserving by adding an LDP mechanism on the gradient-Lipschitz constants. We define this mechanism below and refer to it as the *Gamma mechanism*.

Definition 2 (Gamma Mechanism): The Gamma mechanism takes as input the gradient-Lipschitz constant L_i of node i and outputs

$$\mathcal{R}(L_i) \sim Gamma\left(\frac{L_i}{\theta},\, \theta\right),$$

where $\theta > 0$ is a noise parameter that controls the privacy level.

As a reminder, the probability density function of a random variable L distributed according to Gamma(k, θ) with a shape $k > 0$ and scale $\theta > 0$ is

$$p_{G,L}(l) = \frac{1}{\Gamma(k)\theta^k} l^{k-1} e^{-\frac{l}{\theta}}, \text{ for } l > 0,$$

where $\Gamma(z) = \int_0^\infty u^{z-1}e^{-u}du$ for $z > 0$.

If k is an integer, the Gamma distribution can be interpreted as the sum of k exponential random variable with mean θ [42]. Computing the mean and the variance of $\mathcal{R}(L_i)$, we get the mean $\mathbb{E}_{\mathcal{R}}[\mathcal{R}(L_i)] = \frac{L_i}{\theta}\theta = L_i$, and the variance $v_{\mathcal{R}}[\mathcal{R}(L_i)] = L_i\theta$ and thus θ determines the noise level in the Gamma mechanism.

Accordingly, we obtain Algorithm 3 that we call Private Weighted Random Walk SGD:

Algorithm 3 Private Weighted Random Walk SGD

Initialization: Initial node $i^{(1)}$ chosen uniformly at random from $[N]$, Initial model $w^{(1)}$ chosen uniformly at random from $\mathcal{W}$.
Every node i computes $\mathcal{R}(L_i)$ and shares it along with $\deg(i)$ with its neighbors.
for $k = 1$ **to** T **do**
  $\hat{\nabla} f\big(w^{(k)}\big) = \frac{\bar{L}}{L_{i^{(k)}}} \nabla f_{i^{(k)}}\big(w^{(k)}\big)$
  $w^{(k+1)} = \Pi_{\mathcal{W}}\Big(w^{(k)} - \gamma^{(k)}\hat{\nabla} f\big(w^{(k)}\big)\Big)$
  Choose node j uniformly at random from $\mathcal{N}\big(i^{(k)}\big)$.
  Generate $p \sim U(0, 1)$ where U is the uniform distribution.
  if $p \leq \min\left\{1,\ \frac{\mathcal{R}(L_j)}{\mathcal{R}(L_{i^{(k)}})}\frac{\deg(i^{(k)})}{\deg(j)}\right\}$ **then**
    $i^{(k+1)} \leftarrow j$
  else
    $i^{(k+1)} \leftarrow i^{(k)}$
  end if
end for
Return: $w^{(T)}$.

In this algorithm, each node i applies the Gamma mechanism to its constant L_i only once and shares $\mathcal{R}(L_i)$ with its neighbors before the start of the random walk. These fixed values $\mathcal{R}(L_i)$'s are then used throughout the algorithm.[3] The stationary distribution of the weighted random walk of Algorithm 3 with the noisy values $\mathcal{R}(L_i)$ is

$$\pi_{w,\mathcal{R}}(i) = \frac{\mathcal{R}\ (L_i)}{\sum_j \mathcal{R}\big(L_j\big)},$$

and it follows the Beta distribution [42], i.e., $\pi_{w,\mathcal{R}}(i) \sim \text{Beta}(\frac{L_i}{\theta},\ \frac{\sum_{j\neq i} L_j}{\theta})$. Note that the probability density function of

[1] The upper bound on the number of activated links stems from the fact that in Algorithms 1 and 2 a node will probabilistically choose to perform multiple updates on its data in consecutive rounds without exchanging information with its neighbours, making these rounds communication-free.

[2] Privacy mechanisms for hiding the model, through adding noise or quantization, can be used in conjunction with the mechanism we propose here for protecting the gradient-Lipschitz constants. However, this is not the focus of this article and we defer the analysis of such schemes to a future work.

[3] Note that the alternative option of generating and sharing a new value $\mathcal{R}(L_i)$ every time node i is activated is vulnerable to an averaging attack that can be implemented by the neighbors of i.

the Beta distribution of a random variable Π, with parameters $\alpha,\ \beta > 0$ is

$$p_{B,\Pi}(\pi) = \frac{\pi^{\alpha-1}(1-\pi)^{\beta-1}}{B(\alpha,\ \beta)}, \quad \text{for}\, \pi \in [0,\ 1],$$

where $B(\alpha,\ \beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$.

Therefore, we get that the mean of $\pi_{w,\mathcal{R}}$ over the privatization randomness is the desired stationary distribution of importance sampling, i.e.,

$$\mathbb{E}_{\mathcal{R}}\big[\pi_{w,\mathcal{R}}\big] = \pi_w(i), \quad \forall i \in V.$$

We can see that the Beta distribution is well fitted to model the probability distribution of a probability [43] where the expected values of the stationary probabilities sums 1, which justifies our choice for the Gamma noise over other traditional additive privacy schemes (e.g., Gaussian, Laplace) since it implies a Beta distribution on the ratio representing the stationary distribution.

Moreover, this implies that the gradient estimate is unbiased at the stationary, that is

$$\mathbb{E}_{\mathcal{R}}\mathbb{E}_{\pi_{w,\mathcal{R}}}\Big[\hat{\nabla} f_i \,|\, \mathcal{R}\big(L_j\big),\, j = 1, \ldots,\, N\Big] = \nabla f,$$

the proof of which can be found in Appendix C.

The next Lemma states the tradeoff between the privacy level, quantified through ϵ and δ, and the noise level represented by θ for the Gamma mechanism.

Lemma 1: Given $\epsilon \geq 0$, the Gamma mechanism is $(\epsilon,\ \delta)$ local differential private with parameter θ satisfying

$$\delta \geq \max\left\{1 - \frac{IG\left(\frac{\sup L}{\theta},\, \left(e^{\epsilon}\frac{\Gamma(\sup L/\theta)}{\Gamma(\inf L/\theta)}\right)^{\frac{\theta}{\sup L - \inf L}}\right)}{\Gamma(\sup L/\theta)},\right.$$
$$\left.\frac{IG\left(\frac{\inf L}{\theta},\, \left(\frac{1}{e^{\epsilon}}\frac{\Gamma(\sup L/\theta)}{\Gamma(\inf L/\theta)}\right)^{\frac{\theta}{\sup L - \inf L}}\right)}{\Gamma(\inf L/\theta)}\right\}$$

where $IG(s,t) \coloneqq \int_0^t u^{s-1}e^{-u}du$ is the incomplete gamma function.

The proof of Lemma 1 is in Appendix D. In Fig. 1 shows plots of the tradeoff among the three parameters ϵ, δ and θ descrbied in the above lemma.

Convergence Analysis: The proposed Gamma mechanism applies the Gamma noise to the gradient-Lipschitz constants L_i's only one time before the start of the algorithm. Therefore, finding the expression of the rate of convergence in Algorithm 3, given the generated noisy Lipschitz constants $\mathcal{R}(L_i)$, follows similar steps as the analysis of non-private version of the algorithm and gives the following bound.

$$\mathbb{E}\Big[f\Big(\bar{w}^{(k)}\Big) - f\big(w^*\big)|\mathcal{R}(L_i),\ i = 1, \ldots, N\Big] = \mathcal{O}\left(\frac{\max\left\{\frac{\overline{\mathcal{R}(L)}}{\inf \mathcal{R}(L)}\sigma^2,\, \big(\overline{\mathcal{R}(L)}\big)^2,\, \frac{1}{\ln 1/\lambda_{P_{w,\mathcal{R}}}}\right\}}{k^{1-q}}\right). \quad (10)$$

To characterize the mean rate of convergence under the Gamma mechanism we need to average the bound above over the noise introduced by the mechanism. This involves finding the mean of the different terms in the right-hand-side of (10), which is not straightforward given that the noise introduced by the Gamma mechanism is not additive. Instead, we analysis a modification of the mechanism that we call *Truncated Gamma* mechanism, which first implements the Gamma mechanism and then truncates the values $\mathcal{R}(L_i)$ to a minimum and maximum constants L_{min} and L_{max} $(L_{min} < L_{max})$. By [24, Proposition 2.1] on the post-processing differential privacy immunity rule, we know that the Truncated Gamma mechanism achieves the privacy guarantees (ϵ, δ) of the non-truncated version and is more amenable to analysis as done below (as the expense of the tightness of the bound):

$$\mathbb{E}_{\mathcal{R}}\Big[\mathbb{E}\Big[f\Big(\bar{w}^{(k)}\Big) - f\big(w^*\big)\Big]\Big] = O\left(\frac{\frac{L_{max}}{L_{min}}\sigma^2 + (L_{\max})^2}{k^{1-q}}\right). \quad (11)$$

We can see that the bound in (11) benefits from the same asymptotic rate of convergence as the non-private version.

V. Numerical Results

In this section, we present our numerical results applying Algorithms 1–3 to a decentralized logistic regression problem and comparing it to asynchronous Gossip SGD algorithm [1]. We compare the Random Walk algorithms to the asynchronous Gossip to show the convergence speed-up with respect to the number of performed SGD iterations in the network. For fair comparison in terms of computation and communication, we assume an asynchronous Gossip algorithm where only one communication link gets activated at every iteration and the model on both side gets updated, exchanged and averaged. Also, we run Algorithm 3 using the LDP privatization scheme based on the Gamma mechanism, and we compare it to the additive Laplace mechanism widely used in the literature [24].

We take the graph $\mathcal{G}$ to be an Erdős-Rényi graph on $N = 100$ nodes with probability of connectivity $p = 0.3$. We assume that the labels $y_i \in \{-1, +1\}$. For the data with label $y_i = 1$, we sample x_i from $\mathcal{N}(\mu, v\,\mathbb{I}_d)$ where d is the feature dimension, μ is the mean, v is the variance and $\mathbb{I}_d$ is the identity matrix of dimension d. For label $y_i = -1$, x_i is sampled from $\mathcal{N}(-\mu, v\,\mathbb{I}_d)$.

Loss Function: The averaged regularized cost function for logistic regression on the distributed data can be expressed as follow:

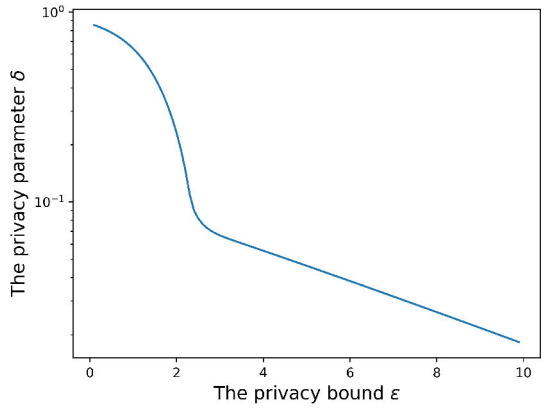
$$f(w) = \sum\nolimits_{i=1}^{N} \log\big(1 + \exp\big(-y_i x_i^T w\big)\big) + \frac{1}{2}\|w\|^2. \quad (12)$$

So, the local loss function at node i is

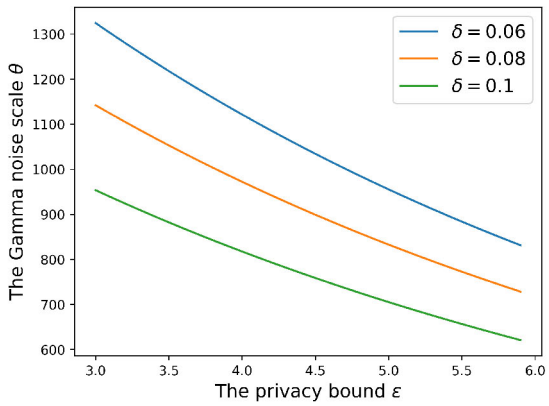
$$f_i(w) = N\log\big(1 + \exp\big(-y_i x_i^T w\big)\big) + \frac{N}{2}\|w\|^2. \quad (13)$$

Then, for such local loss function the gradient Lipschitz constant is $L_i = 1 + \frac{1}{4}N\|x_i\|_2^2$.

Random Walk Algorithms vs the Gossip Algorithm: We consider two data regimes for our simulations: High variance and Low variance regimes. For both regimes, the random walk based algorithms outperform the Gossip algorithm as shown

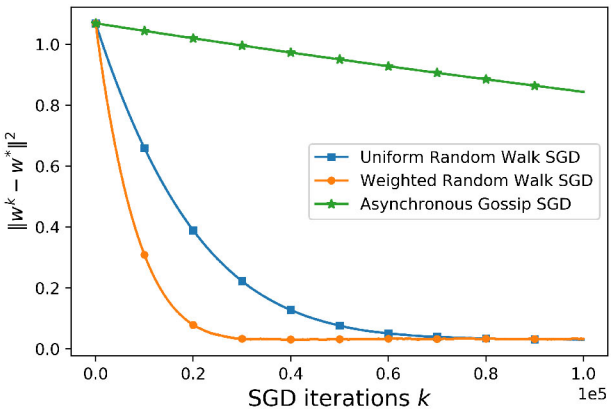


(a) The achievable privacy parameter δ as a function of the privacy bound ϵ.

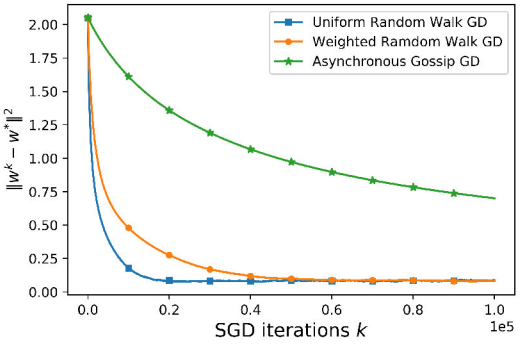


(b) The Gamma noise parameter θ vs. the privacy bound ϵ for $\delta = 0.06,\ 0.08\ \&\ 0.1$.

Fig. 1. The tradeoffs stated in Lemma 1 between the privacy level, quantified through ϵ and δ, and the noise level represented by θ for the Gamma mechanism.



(a) High variance data regime: $v = 10$.



(b) Low variance data regime: $v = 1$.

Fig. 2. Error on the model function of the SGD updates estimated by the Uniform Random Walk SGD (Algorithm 1), Weighted Random Walk SGD (Algorithm 2) and the asynchronous Gossip SGD on an Erdős-Rényi graph of 100 nodes and probability of connectivity $p = 0.3$.

in Figure 2. For the high variance data regime where $v = 10$, the term $\bar{L}$ in the convergence rate dominates, and the weighted sampling gives better convergence. While in the low variance data regime where $v = 1$, the Uniform Random Walk SGD is outperforming the Weighted Random Walk SGD.

Private Random Walk Algorithms: For high variance regime, we simulate the convergence of Algorithm 3 that used the

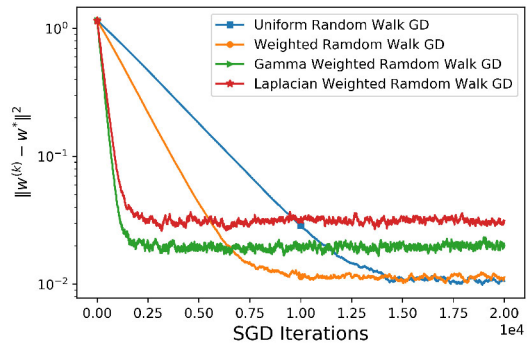


Fig. 3. Error of the model estimated by the Uniform Random Walk SGD, (3, 0.03)-LDP Gamma Weighted Random walk (Algorithm 3) and (3, 0)-LDP Laplace Weighted Random walk on an Erdős-Rényi graph of 20 nodes and probability of connectivity $p = 0.3$.

Gamma mechanism. We compare its performance to a variant that uses an additive Laplacian noise, [24] for $\epsilon = 3$ and $\delta \leq 0.03$ with same noise variance as the Gamma. Our results show that the Gamma mechanism outperforms the Laplacian mechanism .

VI. Conclusion

In this article, we study a decentralized learning problem where the data is distributed over the nodes of a graph. We propose decentralized learning algorithms based on random walk. To speed up the convergence, we propose a weighted random walk algorithm where the nodes get sampled depending on the importance of their data measured through the gradient-Lipschitz constant. A weighted random walk requires every node to share its gradient-Lipschitz constant with the neighbor nodes, which creates a privacy vulnarability. We propose a local differential privacy mechanism that we call Gamma mechanism to address the privacy concern, and give the trade-off between the privacy parameters and the Gamma noise parameter. We also presented numerical results on the convergence of the proposed algorithms. As for future work, we think it is interesting to investigate different importance sampling measures for the weighted random walk.

APPENDIX A
PROOF OF THEOREM 1: CONVERGENCE RATE OF ALGORITHM 1

First, we present preliminary results that will be used in the main proof later.

Lemma 2 (Lipschitzness): If f_i is a convex function on an open subset $\Omega \subseteq \mathbb{R}$, then for a closed bounded subset $\mathcal{W} \subset \Omega$, there exists a constant $D_i \geq 0$, such that, for any $w_1, w_2 \in \mathcal{W}$,

$$|f_i(w_1) - f_i(w_2)| \leq D_i \|w_1 - w_2\|_2.$$

We define $D = \sup_{i \in V} D_i$. Therefore,

$$|f_i(w_1) - f_i(w_2)| \leq D \|w_1 - w_2\|_2.$$

A proof for Lemma 2 can be found in [44].

Corollary 1 (Boundedness of the Gradient): If f_i is a convex function on $\mathbb{R}$, then for a closed bounded subset $\mathcal{W} \subset \mathbb{R}$, $\|\nabla f_i(w)\|_2 \leq D, \quad \forall w \in \mathcal{W}$.

Proof: Taking $v = w + \nabla f_i(w)$,

$$\begin{aligned} D\|\nabla f_i(w)\|_2 &= D\|v - w\|_2 \\ &\overset{(a)}{\geq} |f_i(v) - f_i(w)| \\ &\overset{(b)}{\geq} \langle \nabla f_i(w), \nabla f_i(w) \rangle \\ &= \|\nabla f_i(w)\|_2^2. \end{aligned}$$

(a) follows Lemma 2 and (b) follows from f_i being convex. ■

Next we give some results we need on the Markov chain. We remind the reader that π is the stationary distribution, P is the transition matrix and P^k is the k^{th} power of matrix P. We refer to i^{th} row of a matrix P by $P(i, :)$.

Lemma 3 (Convergence of Markov Chain [26]): Under Assumption 2, we have

$$\max_i \left\| \pi - P^k(i, :) \right\| \leq C \lambda_P^k$$

for $k > K_P$, where K_P is a constant that depends on λ_P and $\lambda_2(P)$ and C is a constant that depends on the Jordan canonical form of P.

Next, we start the proof for Theorem 1 following the same reasoning in [2] but with the inclusion of the gradient-Lipschitz assumption as in Assumption 1.

$$\begin{aligned} \left\| w^{(k+1)} - w^* \right\|_2^2 &= \left\| \mathbf{\Pi}_{\mathcal{W}} \left(w^{(k)} - \gamma^{(k)} \nabla f_{i^{(k)}} \left(w^{(k)} \right) \right) \right. \\ &\qquad \left. - \mathbf{\Pi}_{\mathcal{W}} \left(w^* \right) \right\|_2^2 \\ &\overset{(a)}{\leq} \left\| w^{(k)} - \gamma^{(k)} \nabla f_{i^{(k)}} \left(w^{(k)} \right) - w^* \right\|_2^2 \\ &= \left\| w^{(k)} - w^* \right\|_2^2 \\ &\quad - 2\gamma^{(k)} \left\langle w^{(k)} - w^*, \nabla f_{i^{(k)}} \left(w^{(k)} \right) \right\rangle \\ &\quad + \left(\gamma^{(k)} \right)^2 \left\| \nabla f_{i^{(k)}} \left(w^{(k)} \right) \right\|_2^2 \\ &= \left\| w^{(k)} - w^* \right\|_2^2 \\ &\quad - 2\gamma^{(k)} \left\langle w^{(k)} - w^*, \nabla f_{i^{(k)}} \left(w^{(k)} \right) \right\rangle \\ &\quad + \left(\gamma^{(k)} \right)^2 \left\| \nabla f_{i^{(k)}} \left(w^{(k)} \right) - \nabla f_{i^{(k)}} \left(w^* \right) \right. \\ &\qquad \left. + \nabla f_{i^{(k)}} \left(w^* \right) \right\|_2^2 \\ &\overset{(b)}{\leq} \left\| w^{(k)} - w^* \right\|_2^2 \\ &\quad - 2\gamma^{(k)} \left\langle w^{(k)} - w^*, \nabla f_{i^{(k)}} \left(w^{(k)} \right) \right\rangle \\ &\quad + 2(\gamma^{(k)})^2 \left\| \nabla f_{i^{(k)}} \left(w^{(k)} \right) - \nabla f_{i^{(k)}} \left(w^* \right) \right\|_2^2 \\ &\quad + 2(\gamma^{(k)})^2 \left\| \nabla f_{i^{(k)}} \left(w^* \right) \right\|_2^2. \end{aligned} \tag{14}$$

(a) follows from $\mathcal{W}$ being a convex closed set, so one can apply nonexpansivity theorem [45, Fact E.9.0.0.5], (b) follows from Jensen's inequality applied to the squared norm.

Using Lemma 2 and the convexity of the functions f_i, we get

$$\begin{aligned} \left\| w^{(k+1)} - w^* \right\|_2^2 &\overset{(a)}{\leq} \left\| w^{(k)} - w^* \right\|_2^2 \\ &\quad - 2\gamma^{(k)} \left\langle w^{(k)} - w^*, \nabla f_{i^{(k)}} \left(w^{(k)} \right) \right\rangle \\ &\quad + 2\left(\gamma^{(k)} \right)^2 L_{i^{(k)}}^2 \left\| w^{(k)} - w^* \right\|_2^2 \\ &\quad + 2\left(\gamma^{(k)} \right)^2 \left\| \nabla f_{i^{(k)}} \left(w^* \right) \right\|_2^2 \\ &\overset{(b)}{\leq} \left\| w^{(k)} - w^* \right\|_2^2 \\ &\quad - 2\gamma^{(k)} \left\langle w^{(k)} - w^*, \nabla f_{i^{(k)}} \left(w^{(k)} \right) \right\rangle \\ &\quad + 2(\gamma^{(k)})^2 (\sup L)^2 \left\| w^{(k)} - w^* \right\|_2^2 \\ &\quad + 2\left(\gamma^{(k)} \right)^2 \left\| \nabla f_i \left(w^* \right) \right\|_2^2. \end{aligned} \tag{15}$$

(a) follows from the Lipschitzness Lemma, (b) follows by bounding by the $\sup L$.

For the next we use the convexity of f_i,

$$\begin{aligned} \left\| w^{(k+1)} - w^* \right\|_2^2 &\leq \left\| w^{(k)} - w^* \right\|_2^2 \\ &\quad - 2\gamma^{(k)} \left(f_{i^{(k)}} \left(w^{(k)} \right) - f_{i^{(k)}} \left(w^* \right) \right) \\ &\quad + 2\left(\gamma^{(k)} \right)^2 (\sup L)^2 \left\| w^{(k)} - w^* \right\|_2^2 \\ &\quad + 2\left(\gamma^{(k)} \right)^2 \left\| \nabla f_{i^{(k)}} \left(w^* \right) \right\|_2^2. \end{aligned} \tag{16}$$

By re-arranging (16), we come to

$$\begin{aligned} &\gamma^{(k)} \left(f_{i^{(k)}} \left(w^{(k)} \right) - f_{i^{(k)}} \left(w^* \right) \right) \\ &\leq \frac{1}{2} \left(\left\| w^{(k)} - w^* \right\|_2^2 - \left\| w^{(k+1)} - w^* \right\|_2^2 \right) \\ &\quad + \left(\gamma^{(k)} \right)^2 (\sup L)^2 \left\| w^{(k)} - w^* \right\|_2^2 \\ &\quad + \left(\gamma^{(k)} \right)^2 \left\| \nabla f_{i^{(k)}} \left(w^* \right) \right\|_2^2. \end{aligned} \tag{17}$$

Now summing (17) over k and using Assumption 1 and the boundedness of $\mathcal{W}$,

$$\begin{aligned} &\sum_k \gamma^{(k)} \left(f_{i^{(k)}} \left(w^{(k)} \right) - f_{i^{(k)}} \left(w^* \right) \right) \\ &\leq \frac{1}{2} \left\| w^{(0)} - w^* \right\|_2^2 + (\sup L)^2 \sum_k \left(\gamma^{(k)} \right)^2 \left\| w^{(k)} - w^* \right\|_2^2 \\ &\quad + \sum_k \left(\gamma^{(k)} \right)^2 \left\| \nabla f_{i^{(k)}} \left(w^* \right) \right\|_2^2 < \infty. \end{aligned} \tag{18}$$

We can see that the previous result shows the dependency on $\sup L$.

$$\begin{aligned}
&\gamma^{(k)}\mathbb{E}\Big[f_{j^{(k)}}\Big(w^{(k-T^{(k)})}\Big)-f_{j^{(k)}}\Big(w^{(k)}\Big)\Big]\\
&\overset{(a)}{\leq} D\gamma^{(k)}\mathbb{E}\Big\|w^{(k-T^{(k)})}-w^{(k)}\Big\|\\
&\overset{(b)}{\leq} D\gamma^{(k)}\mathbb{E}\left(\sum_{n=k-T^{(k)}}^{k-1}\Big\|w^{(n+1)}-w^{(n)}\Big\|\right)\\
&\overset{(c)}{\leq} D\gamma^{(k)}\sum_{n=k-T^{(k)}}^{k-1}\mathbb{E}\Big(\Big\|w^{(n+1)}-w^{(n)}\Big\|\Big)\\
&\overset{(d)}{\leq} D^2\gamma^{(k)}\sum_{n=k-T^{(k)}}^{k-1}\gamma^{(n)}\\
&\overset{(e)}{\leq} \frac{D^2}{2}\sum_{n=k-T^{(k)}}^{k-1}\left(\Big(\gamma^{(n)}\Big)^2+\Big(\gamma^{(k)}\Big)^2\right)\\
&\leq \frac{D^2}{2}T^{(k)}\Big(\gamma^{(k)}\Big)^2+\frac{D^2}{2}\sum_{n=k-T_k}^{k-1}\Big(\gamma^{(n)}\Big)^2.
\end{aligned}$$

(a) follows from Lemma 2, (b) using triangle inequality, (c) using linearity of expectation, (d) follows Corollary 1 and (e) follows from the Cauchy–Schwarz inequality.

Now taking the summation over k:

$$\begin{aligned}
&\sum_k \gamma^{(k)}\mathbb{E}\Big[f_{j^{(k)}}\Big(w^{(k-T^{(k)})}\Big)-f_{j^{(k)}}\Big(w^{(k)}\Big)\Big]\\
&\quad\leq \sum_k \frac{D^2}{2}T_k\Big(\gamma^{(k)}\Big)^2+\frac{D^2}{2}\sum_k\sum_{n=k-T_k}^{k-1}\Big(\gamma^{(n)}\Big)^2. \quad (19)
\end{aligned}$$

By simply using the assumption on the step size summability, the result is as follows:

$$\begin{aligned}
&\sum_{k=K}^{\infty}\sum_{n=k-T^{(k)}}^{k-1}\Big(\gamma^{(n)}\Big)^2\\
&\quad\leq \sum_{k=K}^{\infty}T^{(k)}\Big(\gamma^{(k)}\Big)^2\leq\frac{2}{\ln(1/\lambda_P)}\sum_{k=K}^{\infty}\ln k.\Big(\gamma^{(k)}\Big)^2<\infty.\\
&\mathbb{E}_{j^{(k)}}\Big[f_{j^{(k)}}\Big(w^{(k-T^{(k)})}\Big)-f_{j^{(k)}}\big(w^*\big)|X_0,X_1,\ldots,X_{k-T^{(k)}}\Big]\\
&\quad=\sum_{i=1}^{N}\Big(f_i\Big(w^{(k-T^{(k)})}\Big)-f_i\big(w^*\big)\Big)\\
&\qquad\times P\Big(j^{(k)}=i|X_0,X_1,\ldots,X_{k-T^{(k)}}\Big)\\
&\quad\overset{(a)}{=}\sum_{i=1}^{N}\Big(f_i\Big(w^{(k-T^{(k)})}\Big)-f_i\big(w^*\big)\Big)P\Big(j^{(k)}=i|X_{k-T^{(k)}}\Big)\\
&\quad=\sum_{i=1}^{N}\Big(f_i\Big(w^{(k-T^{(k)})}\Big)-f_i\big(w^*\big)\Big)P^{t_k}\Big[X_{k-T^{(k)}}|j^{(k)}=i\Big]\\
&\quad\overset{(b)}{\geq}\Big(f\Big(w^{(k-T^{(k)})}\Big)-f\big(w^*\big)\Big)-\frac{N}{2k}. \quad (20)
\end{aligned}$$

(a) using Markov property and (b) using [2, Lemma 1].

Next, we get a bound on

$$\begin{aligned}
&\sum_k \gamma^{(k)}\mathbb{E}\Big[f\Big(w^{(k)}\Big)-f\Big(w^{(k-T^{(k)})}\Big)\Big].\\
&\gamma^{(k)}\mathbb{E}\Big[f\Big(w^{(k-T_k)}\Big)-f\Big(w^{(k)}\Big)\Big]\\
&\overset{(a)}{\leq} ND\gamma^{(k)}\mathbb{E}\Big\|w^{(k-T_k)}-w^{(k)}\Big\|\\
&\overset{(b)}{\leq} ND\gamma^{(k)}\mathbb{E}\left(\sum_{n=k-T_k}^{k-1}\Big\|w^{(n+1)}-w^{(n)}\Big\|\right)\\
&\overset{(c)}{\leq} ND\gamma^{(k)}\sum_{n=k-T_k}^{k-1}\mathbb{E}\Big(\Big\|w^{(n+1)}-w^{(n)}\Big\|\Big)\\
&\overset{(d)}{\leq} ND^2\gamma^{(k)}\sum_{n=k-T_k}^{k-1}\gamma^{(n)}\\
&\overset{(e)}{\leq} \frac{ND^2}{2}\sum_{n=k-T_k}^{k-1}\left(\Big(\gamma^{(n)}\Big)^2+\Big(\gamma^{(k)}\Big)^2\right)\\
&\leq \frac{ND^2}{2}T_k\Big(\gamma^{(k)}\Big)^2+\frac{ND^2}{2}\sum_{n=k-T_k}^{k-1}\Big(\gamma^{(n)}\Big)^2.
\end{aligned}$$

(a) follows from Lemma 4, (b) using triangle inequality, (c) using linearity of expectation, (d) follows Corollary 1 and (e) follows from the Cauchy–Schwarz inequality. The upper bound summability over k follows from previous discussion in equation (20).

Combining with the results in (18) and (20) , we get

$$\begin{aligned}
\sum_k \gamma^{(k)}\mathbb{E}\Big[f\Big(w^{(k-T_k)}\Big)-f\big(w^*\big)\Big]&\leq C_1\sigma^2+C_2(\sup L)^2\\
&\quad+C_3\left(\frac{1}{\ln(1/\lambda_P)}\right).
\end{aligned}$$

Finally,

$$\begin{aligned}
\mathbb{E}\Big(f\Big(\bar{w}^{(k)}\Big)-f\big(w^*\big)\Big)&=O\left(\frac{\max\Big(\sigma^2,\ (\sup L)^2,\ \frac{1}{\ln(1/\lambda_P)}\Big)}{\Big(\sum_{j=1}^{k}\gamma^{(k)}\Big)}\right)\\
&=O\left(\frac{\max\Big(\sigma^2,\ (\sup L)^2,\ \frac{1}{\ln(1/\lambda_P)}\Big)}{k^{1-q}}\right). \quad (21)
\end{aligned}$$

APPENDIX B
PROOF OF THEOREM 2

To prove Theorem 2, we scale the local losses $f_i(w)$ in order to keep the same global loss under the new weighted sampling average is

$$f_{i,w}(w)=\frac{f_i(w)}{\frac{L_i}{\bar{L}}}=\frac{\bar{L}}{L_i}f_i(w). \quad (22)$$

For the next, we compute the gradient Lipschitz constant for the weighted loss function which is $L_{i,w}=\frac{L_i}{\frac{L_i}{\bar{L}}}=\bar{L}$.

Here, we get $\sup L_w := \max_i L_{i,w} = \bar{L}$. After we compute the new residual quantity that also contributes to the rate of convergence:

$$\sum_{i=1}^{N} \frac{L_i}{N\bar{L}} \left\| \nabla f_{i,w}(w^*) \right\|_2^2 = \sum_{i=1}^{N} \frac{L_i}{N\bar{L}} \left(\frac{\bar{L}}{L_i} \right)^2 \left\| \nabla f_i(w^*) \right\|_2^2 = \sum_{i=1}^{N} \frac{1}{N} \left(\frac{\bar{L}}{L_i} \right) \left\| \nabla f_i(w^*) \right\|_2^2 \leq \frac{\bar{L}}{\inf L} \left(\sum_{i=1}^{N} \frac{1}{N} \left\| \nabla f_i(w^*) \right\|_2^2 \right).$$

Appendix C
Private Random Walk SGD

Using the Gamma noise on the gradient-Lipschitz constants in the weighted random walk ends up giving unbiased gradient estimate at the stationary regime:

$$\mathbb{E}_{\mathcal{R}} \mathbb{E}_{\pi_{w,\mathcal{R}}} \left[\hat{\nabla} f_i | \mathcal{R}(L_j),\ j = 1, \ldots, N \right] = \nabla f.$$

Proof:

$$\begin{aligned} &\mathbb{E}_{\mathcal{R}} \mathbb{E}_{\pi_{w,\mathcal{R}}} \left[\hat{\nabla} f_i | \mathcal{R}(L_j),\ j = 1, \ldots, N \right] \\ &= \mathbb{E}_{\mathcal{R}} \mathbb{E}_{\pi_{w,\mathcal{R}}} \left[\frac{\bar{L}}{L_i} \nabla f_i | \mathcal{R}(L_j),\ j = 1, \ldots, N \right] \\ &= \sum_i \mathbb{E}_{\mathcal{R}} \left[\frac{\mathcal{R}(L_i)}{\sum_j \mathcal{R}(L_j)} \frac{\sum_j L_j}{N L_i} \nabla f_i \right] \\ &= \mathbb{E}_i \left[\nabla f_i \right] = \nabla f. \end{aligned}$$

■

Appendix D
Proof of Lemma 1

We will use the following property for (ϵ, δ) differentially mechanism.

Lemma 4 [25]: A mechanism is (ϵ, δ)-differentially private if, for any L, L', and Z a random variable with the same distribution as $\mathcal{R}(L)$, we have

$$P\left(\frac{P_{\mathcal{R}(L)}(z)}{P_{\mathcal{R}(L')}(z)} \geq \epsilon \right) \leq \delta.$$

In our case, $Z \sim Gamma(L, \theta)$.

Now, for the proof of Lemma 1, taking L, L' and a received output $z \in \mathbb{R}$, we have

$$\frac{P_{\mathcal{R}(L)}(z)}{P_{\mathcal{R}(L')}(z)} = \frac{\frac{1}{\Gamma(L/\theta)\theta^{\frac{L}{\theta}}} z^{\frac{L}{\theta}-1} e^{\frac{z}{\theta}}}{\frac{1}{\Gamma(L'/\theta)\theta^{\frac{L'}{\theta}}} z^{\frac{L'}{\theta}-1} e^{\frac{z}{\theta}}} = \frac{\Gamma(L'/\theta)}{\Gamma(L/\theta)} \theta^{\frac{L'-L}{\theta}} z^{\frac{L-L'}{\theta}}.$$

Then, taking a random variable Z on z, where $Z \sim Gamma(L,\ \theta)$:

$$\left(\frac{Z}{\theta} \right)^{\frac{L-L'}{\theta}} \sim Generalized\ Gamma \times \left(p = \frac{\theta}{L - L'},\ d = \frac{L}{L - L'},\ 1 \right),$$

when $L > L'$.

As a reminder, the probability density function of the Generalized Gamma function on a random variable Y is

$$p_{GG,Y}(y) = \frac{p/a^d}{\Gamma(d/p)} y^{d-1} e^{-(y/a)^p} \quad y > 0, \quad a, d, p > 0.$$

When $L > L'$, we have

$$\begin{aligned} P\left(\frac{\Gamma(L'/\theta)}{\Gamma(L/\theta)} \theta^{\frac{L'-L}{\theta}} Z^{\frac{L-L'}{\theta}} \geq e^{\epsilon} \right) &= P\left(\left(\frac{Z}{\theta} \right)^{\frac{L-L'}{\theta}} \geq e^{\epsilon} \frac{\Gamma(L/\theta)}{\Gamma(L'/\theta)} \right) \\ &= 1 - P\left(\left(\frac{Z}{\theta} \right)^{\frac{L-L'}{\theta}} \leq e^{\epsilon} \frac{\Gamma(L/\theta)}{\Gamma(L'/\theta)} \right) \\ &= 1 - \frac{IG\left(\frac{L}{\theta}, \left(e^{\epsilon} \frac{\Gamma(L/\theta)}{\Gamma(L'/\theta)} \right)^{\frac{\theta}{L'-L}} \right)}{\Gamma(L/\theta)} \\ &\leq \delta, \end{aligned}$$

where IG is the lower incomplete gamma function.

When $L < L'$, we have

$$\begin{aligned} P\left(\frac{\Gamma(L'/\theta)}{\Gamma(L/\theta)} \theta^{\frac{L'-L}{\theta}} Z^{\frac{L-L'}{\theta}} \geq e^{\epsilon} \right) &= P\left(\left(\frac{Z}{\theta} \right)^{\frac{L'-L}{\theta}} \leq \frac{\Gamma(L'/\theta)}{\Gamma(L/\theta) e^{\epsilon}} \right) \\ &= \frac{IG\left(\frac{L}{\theta}, \left(\frac{\Gamma(L'/\theta)}{\Gamma(L/\theta) e^{\epsilon}} \right)^{\frac{\theta}{L'-L}} \right)}{\Gamma(L/\theta)} \\ &\leq \delta. \end{aligned}$$

References

[1] S. S. Ram, A. Nedic, and V. V. Veeravalli, "Asynchronous gossip algorithms for stochastic optimization," in *Proc. Int. Conf. Game Theory Netw.*, 2009, pp. 80–81.

[2] T. Sun, Y. Sun, and W. Yin, "On Markov chain gradient descent," in *Proc. NeurIPS*, 2018, pp. 9918–9927.

[3] D. Needell, R. Ward, and N. Srebro, "Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Red Hook, NY, USA: Curran Assoc., Inc., 2014, pp. 1017–1025.

[4] B. Johansson, M. Rabi, and M. Johansson, "A simple peer-to-peer algorithm for distributed optimization in sensor networks," in *Proc. 46th IEEE Conf. Decis. Control*, 2007, pp. 4705–4710.

[5] B. Johansson, M. Rabi, and M. Johansson, "A randomized incremental subgradient method for distributed optimization in networked systems," *SIAM J. Optim.*, vol. 20, no. 3, pp. 1157–1170, 2009.

[6] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. Autom. Control*, vol. 54, pp. 48–61, Jan. 2009.

[7] H. Wai, N. Freris, A. Nedic, and A. Scaglione, "SUCAG: Stochastic unbiased curvature-aided gradient method for distributed optimization," in *Proc. IEEE Conf. Decis. Control (CDC)*, 2018, pp. 1751–1756.

[8] H. Wai, W. Shi, A. Nedic, and A. Scaglione, "Curvature-aided incremental aggregated gradient method," in *Proc. 55th Annu. Allerton Conf. Commun. Control Comput. (Allerton)*, 2017, pp. 526–532.

[9] X. Mao, K. Yuan, Y. Hu, Y. Gu, A. H. Sayed, and W. Yin, "Walkman: A communication-efficient random-walk algorithm for decentralized optimization," *IEEE Trans. Signal Process.*, vol. 68, pp. 2513–2528, 2020.

[10] S. P. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2508–2530, Jun. 2006.

[11] A. Koloskova, S. U. Stich, and M. Jaggi, "Decentralized stochastic optimization and gossip algorithms with compressed communication," 2019. [Online]. Available: arXiv:1902.00340.

[12] T. Aysal, M. E. Yildiz, A. Sarwate, and A. Scaglione, "Broadcast gossip algorithms for consensus," *IEEE Trans. Signal Process.*, vol. 57, no. 7, pp. 2748–2761, Jul. 2009.
[13] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Trans. Autom. Control*, vol. 57, no. 3, pp. 592–606, Mar. 2012.
[14] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Local privacy and statistical minimax rates," in *Proc. IEEE 54th Annu. Symp. Found. Comput. Sci.*, 2013, pp. 429–438.
[15] P. Kairouz, S. Oh, and P. Viswanath, "Extremal mechanisms for local differential privacy," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2879–2887.
[16] X. Wu, A. Kumar, K. Chaudhuri, S. Jha, and J. F. Naughton, "Differentially private stochastic gradient descent for in-RDBMS analytics," 2016. [Online]. Available: arXiv:1606.04722.
[17] R. Bassily, A. D. Smith, and A. Thakurta, "Private empirical risk minimization: Efficient algorithms and tight error bounds," in *Proc. IEEE 55th Annu. Symp. Found. Comput. Sci.*, 2014, pp. 464–473.
[18] V. Gandikota, R. K. Maity, and A. Mazumdar, "VQSGD: Vector quantized stochastic gradient descent," 2019. [Online]. Available: arXiv:1911.07971.
[19] S. Xiong, A. D. Sarwate, and N. B. Mandayam, "Randomized requantization with local differential privacy," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2016, pp. 2189–2193.
[20] W.-N. Chen, P. Kairouz, and A. Özgür, "Breaking the communication-privacy-accuracy trilemma," 2020. [Online]. Available: arXiv:2007.11707.
[21] P. Zhao and T. Zhang, "Stochastic optimization with importance sampling," 2014. [Online]. Available: arXiv:1401.2753.
[22] G. Ayache and S. E. Rouayheb, "Random walk gradient descent for decentralized learning on graphs," in *Proc. IEEE Int. Parallel Distrib. Process. Symp. Workshops (IPDPSW)*, 2019, pp. 926–931.
[23] D. P. Bertsekas, "Nonlinear programming," *J. Oper. Res. Soc.*, vol. 48, p. 334, Dec. 1997.
[24] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, nos. 3–4, pp. 211–407, 2014.
[25] M. Bun and T. Steinke, "Concentrated differential privacy: Simplifications, extensions, and lower bounds," 2016. [Online]. Available: arXiv:1605.02065.
[26] D. A. Levin, Y. Peres, and E. L. Wilmer, *Markov Chains and Mixing Times*. Providence, RI, USA: Amer. Math. Soc., 2006.
[27] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Stat.*, vol. 22, no. 3, pp. 400–407, 1951.
[28] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. COMPSTAT*, 2010, pp. 177–186.
[29] F. R. Bach and E. Moulines, "Non-asymptotic analysis of stochastic approximation algorithms for machine learning," in *Proc. NIPS*, 2011, pp. 451–459.
[30] C.-H. Lee, X. Xu, and D. Y. Eun, "Beyond random walk and metropolis-hastings samplers: Why you should not backtrack for unbiased graph sampling," in *Proc. SIGMETRICS*, 2012, pp. 319–330.
[31] B. Chen, Y. Xu, and A. Shrivastava, "Fast and accurate stochastic gradient estimation," in *Proc. NeurIPS*, 2019, pp. 12339–12349.
[32] Z. Borsos, S. Curi, K. Y. Levy, and A. Krause, "Online variance reduction with mixtures," in *Proc. ICML*, 2019, pp. 705–714.
[33] G. Alain, A. Lamb, C. Sankar, A. C. Courville, and Y. Bengio, "Variance reduction in SGD by distributed importance sampling," 2015. [Online]. Available: arXiv:1511.06481.
[34] G. Bouchard, T. Trouillon, J. Perez, and A. Gaidon, "Online learning to sample," 2015. [Online]. Available: arXiv:1506.09016.
[35] L. Bottou and O. Bousquet, "The tradeoffs of large scale learning," in *Proc. NIPS*, 2007, pp. 161–168.
[36] F. R. Bach and E. Moulines, "Non-strongly-convex smooth stochastic approximation with convergence rate O(1/n)," in *Proc. NIPS*, 2013, pp. 773–781.
[37] J. C. Duchi, A. Agarwal, M. Johansson, and M. I. Jordan, "Ergodic mirror descent," in *Proc. 49th Annu. Allerton Conf. Commun. Control Comput. (Allerton)*, 2011, pp. 701–706.
[38] W. Shi, Q. Ling, G. Wu, and W. Yin, "EXTRA: An exact first-order algorithm for decentralized consensus optimization," *SIAM J. Optim.*, vol. 25, no. 2, pp. 944–966, 2015.
[39] A. I. Chen, "Fast distributed first-order methods," M.S. thesis, Dept. Electr. Eng. Comput., Massachusetts Ins. Technol., Cambridge, MA, USA, 2012.
[40] D. Jakovetić, J. Xavier, and J. M. F. Moura, "Fast distributed gradient methods," *IEEE Trans. Autom. Control*, vol. 59, no. 5, pp. 1131–1146, May 2014.
[41] A. Koloskova, N. Loizou, S. Boreiri, M. Jaggi, and S. U. Stich, "A unified theory of decentralized sgd with changing topology and local updates," 2020. [Online]. Available: arXiv:2003.10422.
[42] A. Papoulis and S. U. Pillai, *Probability, Random Variables, and Stochastic Processes*, 4th ed. Boston, MA, USA: McGraw-Hill, 2002.
[43] A. Kim. (Jan. 2020). *Beta Distribution: Intuition, Examples, and Derivation*. Accessed: Jan. 8, 2020. [Online]. Available: https://towardsdatascience.com/beta-distribution-intuition-examples-and-derivation-cf00f4db57af
[44] "Every convex function is locally Lipschitz," *Amer. Math. Monthly*, vol. 79, no. 10, pp. 1121–1124, 1972. [Online]. Available: http://www.jstor.org/stable/2317434
[45] J. C. Dattorro, *Convex Optimization and Euclidean Distance Geometry*, Meboo Publ. USA, Palo Alto, CA, USA, 2004.