Straggler-Resilient Differentially Private Decentralized Learning

Yauhen Yakimenka, *Member, IEEE*, Chung-Wei Weng, Hsuan-Yin Lin, *Senior Member, IEEE*, Eirik Rosnes, *Senior Member, IEEE*, and Jörg Kliewer, *Fellow, IEEE*

***Abstract*—We consider the straggler problem in decentralized learning over a logical ring while preserving user data privacy. Especially, we extend the recently proposed framework of differential privacy (DP) amplification by decentralization by Cyffers and Bellet to include overall training latency—comprising both computation and communication latency. Analytical results on both the convergence speed and the DP level are derived for both a skipping scheme (which ignores the stragglers after a timeout) and a baseline scheme that waits for each node to finish before the training continues. A trade-off between overall training latency, accuracy, and privacy, parameterized by the timeout of the skipping scheme, is identified and empirically validated for logistic regression on a real-world dataset and for image classification using the MNIST and CIFAR-10 datasets.**

***Index Terms*—Decentralized learning, differential privacy, gradient descent, privacy amplification, straggler mitigation, training latency.**

I. Introduction

In distributed learning, a finite-sum optimization problem is solved across multiple nodes without exchanging the local datasets directly, thus improving user data privacy and reducing the communication cost. A popular instance of distributed learning is federated learning [2], [3], [4] in which there is a single central server coordinating the training process. On the other hand, in fully decentralized learning, see, e.g., [5], [6], there is no such coordinating central server—the nodes maintain a local estimate of the optimal model and iteratively update it by averaging estimates obtained from neighbors corrected on the basis of their local datasets. There are two modes of operation—sequential and parallel—and theoretical studies show that the physical communication topology has a strong impact on the number of epochs needed to converge [7].

It is well-known by now that the computed partial (sub)gradients can leak information on the local datasets [8]. In order to circumvent this, a carefully selected noise term can be added to the computed partial (sub)gradients before they are transmitted to other nodes, referred to as local differential privacy (LDP) [9], [10]. In fully decentralized learning, nodes have only a local view of the system. Hence, Cyffers and Bellet [11] recently proposed a novel relaxation of LDP, referred to as network DP (NDP), to naturally capture this. Furthermore, they showed that the privacy-utility trade-off under NDP can be significantly improved upon compared to what is achievable under LDP, illustrating that formal privacy gains can be obtained from full decentralization, complementing previous notions of "amplifying" the privacy by shuffling, subsampling, and iteration [12], [13], [14], [15]. Recently, the work in [11] was extended to a parallel approach that alternates between local gradient descent steps for all nodes in parallel and subsequent gossip averaging [16]. Accordingly, the NDP concept was relaxed to capture that the privacy leakage from a node to another node may depend on their distance in the graph. It was shown in [16] that privacy amplification can be achieved as for the sequential approach in [11]. Differentially-private fully decentralized learning has also been considered in several other previous works, see, e.g., [6], [17], [18]. In the federated learning case, there are numerous works that consider user privacy, e.g., both from a DP perspective (see, e.g., [10]) and from an information-theoretic secure aggregation perspective (see, e.g., [19], [20], [21], [22], [23]).

The problem of *straggling* nodes, i.e., nodes that take a long time to finish their tasks due to random phenomena such as processes running in the background and memory access, has been broadly studied in the literature. The *ignoring-stragglers strategy*, i.e., ignoring results from the slowest nodes, see, e.g., [6], [24], is simple and popular, but can lead to convergence to a local optimum when the data is heterogeneous [25], [26]. Coded computing methods [27], [28], [29] is an alternative to provide resiliency against straggling nodes, and the key idea is to add redundancy to the computation through an error-correcting code. The coded computing literature has considered several different computing tasks, e.g., vector-matrix multiplication [27], [30], [31], (secure) distributed matrix-matrix multiplication [32], [33],

Manuscript received 29 October 2023; revised 22 February 2024; accepted 19 April 2024. Date of publication 20 May 2024; date of current version 1 July 2024. This work was supported by the Experimental Infrastructure for Exploration of Exascale Computing (eX3), which is financially supported by the Research Council of Norway under Contract 270053. The work of Yauhen Yakimenka and Jörg Kliewer was supported in part by U.S. NSF under Grant 1815322, Grant 1908756, and Grant 2107370. This paper was presented in part at the IEEE Information Theory Workshop (ITW), Mumbai, India, November 2022 [DOI: 10.1109/ITW54588.2022.9965898]. *(Corresponding author: Eirik Rosnes.)*

Yauhen Yakimenka and Jörg Kliewer are with the Helen and John C. Hartmann Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA (e-mail: yauhen.yakimenka@njit.edu; jkliewer@njit.edu).

Chung-Wei Weng, Hsuan-Yin Lin, and Eirik Rosnes are with Simula UiB, 5006 Bergen, Norway (e-mail: chungwei@simula.no; lin@simula.no; eirikrosnes@simula.no).

This article has supplementary downloadable material available at https://doi.org/10.1109/JSAIT.2024.3400995, provided by the authors.

Digital Object Identifier 10.1109/JSAIT.2024.3400995

[34], [35], [36], [37], [38], [39], [40], [41], and more general distributed optimization and nonlinear computation problems [42], [43], [44], [45], [46], [47], [48], [49]. For matrix-matrix multiplication, the state-of-the-art for straggler mitigation is achieved by the combination of the results in [47] and [40].

In this work, we study the impact of stragglers and user data privacy in decentralized training. In particular, we assume an underlying physical full mesh topology, i.e., all nodes can physically communicate with each other, but sequential training along a logical ring on top of the physical topology where each node communicates a token only with its immediate neighbors upstream and downstream. In sequential training, nodes do not need to be active during the whole training period, which makes it suitable for scenarios where the nodes have limited resources, and therefore remain dormant unless they are triggered to do an update. See also [50], [51] for further motivation for this scenario. For this setting, we extend the recently proposed framework of privacy amplification by decentralization by Cyffers and Bellet [11] to include the overall latency—comprising both computation and communication latency—under stochastic gradient descent. Our main contributions are summarized as follows.

- We study a skipping scheme (which ignores the stragglers after a timeout) and a baseline scheme that waits for each node to finish its computation before the training continues, for a fixed and a randomized ring topology, and derive analytical results on the convergence behavior (see Theorem 1) and the DP level (see Theorems 2 and 3), revealing a trade-off parameterized by the timeout of the skipping scheme. We show that the asymptotic convergence rate is equal to that of [52, Th. 2]. We note that the presented proofs in Appendices A and B require several nontrivial steps which can not be found in previous work, e.g., the asymptotic convergence analysis in Appendix E in the supplementary material and the adaption to a decreasing learning rate in Appendix B. See also the first paragraph of Section IV. Moreover, we emphasize again that this work studies the effect of stragglers, which by itself is novel for the considered scenario.
- The optimal timeout that minimizes the time between two consecutive updates of the token is determined, showing that skipping is beneficial for faster convergence for certain popular computational delay models considered in the literature (see Lemma 2 and Section VI-C).
- We show that randomizing the processing order of nodes on the ring yields an improvement in both convergence behavior and privacy in the long run (see Section VI-B), although the error and the privacy leakage level show the same order-wise asymptotic behavior in the number of update steps with and without randomization (see Remark 2). This is in particular prominent for a larger number of nodes due to the increased effect of privacy amplification.

Finally, we present extensive empirical results for both logistic regression on a binarized version of the UCI housing dataset [53] and for image classification using both the MNIST [54] and CIFAR-10 [55] datasets to validate our theoretical findings. We also compare with the parallel approach from [16] and to a centralized federated learning approach.[1]

II. Preliminaries

A. Notation

We use uppercase and lowercase letters for random variables (RVs) and their realization (both scalars and vectors), respectively, and italics for sets, e.g., X, x, and $\mathcal{X}$ represent a RV, a scalar/vector, and a set, respectively. An exception to this rule is τ which denotes the model description, also referred to as the token. Matrices are denoted by uppercase letters, their distinction from RVs will be clear from the context. Vectors are represented as row vectors and the transpose of a vector or a matrix is denoted by $(\cdot)^\top$. The expectation of a RV X is denoted by $\mathbb{E}[X]$. We define $[n] \triangleq \{1, 2, \ldots, n\}$, while $\mathbb{N}$ denotes the set of natural numbers and $\mathbb{R}$ the set of real numbers. The (sub)gradient of a function $f(x)$ is denoted by $\nabla f(x)$, while the ℓ_p-norm of a length-n vector $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$ is denoted by $\|x\|_p = (\sum_{i=1}^n |x_i|^p)^{1/p}$, where $|\cdot|$ denotes absolute value. The base of the natural logarithm is denoted by e, while log denotes natural logarithm. $\mathcal{N}(\mu, \sigma^2 I_d)$ denotes the d-dimensional Gaussian (uncorrelated) distribution with mean μ and standard deviation σ of each component, where I_d is the identity matrix of size d. $X \sim \mathcal{P}$ denotes that X is distributed according to the distribution $\mathcal{P}$, while $x \sim \mathcal{P}$ denotes a sample x taken from $\mathcal{P}$. We denote by $\mathcal{D} \sim_u \mathcal{D}'$ the fact that datasets $\mathcal{D} = \cup_{v\in\mathcal{V}}\mathcal{D}_v$ and $\mathcal{D}' = \cup_{v\in\mathcal{V}}\mathcal{D}'_v$ are the same except perhaps for the dataset of user u, i.e., $\mathcal{D}_v = \mathcal{D}'_v$ for all $v \neq u$, where $\mathcal{V}$ is some set of users. Standard order notation $O(\cdot)$ is used for asymptotic results.

B. Definitions and Assumptions

Definition 1 (k-Lipschitz Continuity): A function $f : \mathcal{W} \to \mathbb{R}$ is *k-Lipschitz continuous* over the convex domain $\mathcal{W} \subseteq \mathbb{R}^d$ if $|f(w) - f(w')| \leq k\|w - w'\|_2$ for all $w, w' \in \mathcal{W}$.

Definition 2 (β-Smoothness): A function $f : \mathcal{W} \to \mathbb{R}$ is *β-smooth* over the convex domain $\mathcal{W} \subseteq \mathbb{R}^d$ if $\|\nabla f(w) - \nabla f(w')\|_2 \leq \beta\|w - w'\|_2$ for all $w, w' \in \mathcal{W}$.

Assumption 1: $f_v(\tau; \cdot)$, $v \in \mathcal{V}$, is k-Lipschitz continuous and convex in its first argument.

Assumption 2: f_v, $v \in \mathcal{V}$, is β-smooth.

C. System Model

Consider a decentralized network of n honest-but-curious nodes (users) $\mathcal{V} = \{v_1, \ldots, v_n\}$ with a decentralized dataset $\mathcal{D} = \cup_{v\in\mathcal{V}}\mathcal{D}_v$ where $\mathcal{D}_v = \{(x_i^{(v)}, y_i^{(v)})\}_{i=1}^{\kappa}$, $(x_i^{(v)}, y_i^{(v)}) \in \mathcal{R} \subseteq \mathbb{R}^{d_x} \times \mathbb{R}^{d_y}$, for some set $\mathcal{R}$ and $d_x, d_y, \kappa \in \mathbb{N}$, is the private dataset of node $v \in \mathcal{V}$.

[1] Compared to the conference version [1], we provide a *complete* exposition that includes all technical proofs, as well as new asymptotic results, in addition to significantly extended numerical results. Missing proofs (including the proof of Remark 2) can be found in Appendices D and E in the supplementary material. The code for this work is available at https://github.com/Simula-UiB/SRDPDL_JSAIT24.

The nodes want to compute some function together based on their datasets but want to keep their datasets private. For that, they employ a decentralized protocol where a token $\tau \in \mathcal{W}$, for some convex set $\mathcal{W} \subseteq \mathbb{R}^d$, travels between the nodes according to some predefined (but potentially randomized) path. When receiving the token the r-th time and the global time is h, the node v updates it as $\tau \leftarrow g_r^{(v)}(\tau; \texttt{state}_v(h))$, and sends it further. Here, $\texttt{state}_v(h)$ encapsulates all the information available to the node v at time h, e.g., the available data points and the results of previous calculations. It can also include some source of randomness. We assume that the computation in each node v during the r-th visit of the token takes random time $T_r^{(v)}$. Hence, the computation of $g_r^{(v)}(\cdot, \cdot)$ takes time at most $T_r^{(v)}$ as the token may be updated before the entire computation is finished.[2] We consider a model where $T_r^{(v)}$ is comprised of a deterministic constant part (the time it takes for an actual computation) and a random part. Also, we assume that communication between any two nodes is noiseless and takes constant time χ, and hence the constant part of the computation time can be set to zero. At the end of the protocol, the token τ is distributed among the nodes, which allows for calculating the desired result. This final distribution takes constant overhead time and is therefore ignored.

For a decentralized protocol $\mathcal{A}$, we denote by $\mathcal{A}(\mathcal{D})$ the (random) transcript of all messages sent or received by all the users, i.e., $\mathcal{A}(\mathcal{D})$ are all the triples (u, w, v), if $u \in \mathcal{V}$ sent a message with content w to $v \in \mathcal{V}$. However, due to the decentralized nature of $\mathcal{A}$, the user v only has access to the subset of $\mathcal{A}(\mathcal{D})$ consisting of the messages she sent or received, and we denote this view by $\mathcal{O}_v(\mathcal{A}(\mathcal{D})) = \{(u, w, u') \in \mathcal{A}(\mathcal{D}) : u = v \text{ or } u' = v\}$. Let Ω denote the set of all possible views, i.e., $\mathcal{O}_v(\mathcal{A}(\mathcal{D})) \in \Omega$ for all possible parameters and realizations.

D. Network Differential Privacy

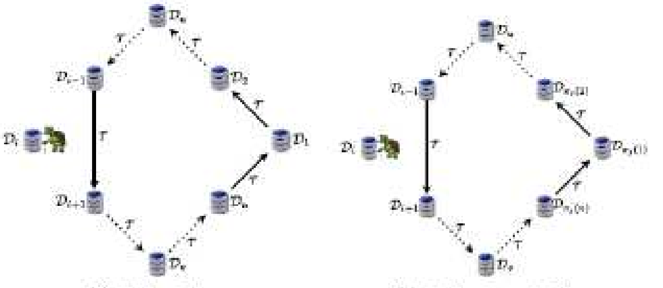
We accept the notion of NDP introduced in [11].

Definition 3 (NDP [11]): A protocol $\mathcal{A}$ satisfies (ε, δ)-NDP if for all pairs of distinct users $u, v \in \mathcal{V}$, all pairs of neighboring datasets $\mathcal{D} \sim_u \mathcal{D}'$, and any $\mathcal{S} \subseteq \Omega$, we have

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

where the notion of neighboring datasets $\mathcal{D} \sim_u \mathcal{D}'$ is defined in Section II-A.

NDP measures how much the information collected by node v depends on the dataset of node u. In the special case that all nodes can observe all messages, i.e., $\mathcal{O}_v$ is the identity map, NDP boils down to conventional LDP [57]. When processing information in a decentralized manner with no central coordinating entity, and when there is no third party (on top of the topology) observing all messages sent, NDP is a more natural privacy measure than DP or LDP.



(a) Skip-Ring. (b) Skip-Rand-Ring.

Fig. 1. Illustrating the j-th round in which node v_i is a straggler.

III. Empirical Risk Minimization

In this section, we consider the empirical risk minimization problem

$$\tau^* = \underset{\tau \in \mathcal{W} \subseteq \mathbb{R}^d}{\arg\min} \left[f(\tau; \mathcal{D}) \triangleq \frac{1}{n} \sum_{v \in \mathcal{V}} f_v(\tau; \mathcal{D}_v) \right], \tag{1}$$

where $f_v(\tau; \cdot)$ is k-Lipschitz continuous and convex in its first argument (see Assumption 1).

A. Skipping Scheme

We suggest the following protocol inspired by projected noisy stochastic gradient descent to solve (1). The token τ keeps the current estimate of the optimal point τ^* and follows a possibly randomized path over the available nodes $\mathcal{V}$. To speed up the process, the token waits up to a threshold time t_{skip} and, if the computation has not finished by that time, the token is forwarded further without an update.[3] In our notation, it means that the calculation in each node v is

$$g_r^{(v)}(\tau; \texttt{state}_v(h)) = \begin{cases} \Pi_{\mathcal{W}}(\tau - \eta_h(\nabla f_v(\tau; \mathcal{D}_v) + N_h)) & \text{if } T_r^{(v)} \leq t_{\text{skip}}, \\ \tau & \text{otherwise}, \end{cases} \tag{2}$$

where η_h is the step size (learning rate), $\Pi_{\mathcal{W}}$ denotes the Euclidean projection onto the set $\mathcal{W}$, and N_h is noise with zero mean and standard deviation σ_h. The noise N_h is added in order to protect the privacy of the local datasets, and the standard deviation σ_h is chosen so a certain level of NDP is ensured.[4] In this work, we consider the gamma distribution (including the exponential distribution) and the Pareto type II (also known as Lomax) distribution for $T_r^{(v)}$, which are well-established models in the literature, see, e.g., [56], [58], [59]. Since we assume that the RVs $T_r^{(v)}$ are i.i.d., we simplify the notation in the following by letting $T \equiv T_r^{(v)}$.

The algorithm stops when a predefined convergence requirement is fulfilled. We refer to the algorithm detailed above as the skipping scheme with parameter t_{skip}, which can be optimized in order to reduce either the convergence time and/or the privacy leakage. In the special case of $t_{\text{skip}} = \infty$,

[2] The RVs $T_r^{(v)}$ are assumed to be independent and identically distributed (i.i.d.) which is in accordance with the literature, where typically stragglers are generated uniformly at random, except for a few works, e.g., [44], [56] that consider a model where nodes tend to remain stragglers for a long time, violating the i.i.d. assumption on the RVs $T_r^{(v)}$.

[3] In practice, acknowledgments can identify straggling nodes: if the token is sent to the next node in line and not acknowledged within a threshold time, it is forwarded to the next node in line, etc.

[4] The noise follows $\mathcal{N}\left(0, \sigma_h^2\right)$, where $\sigma_h = \frac{k\sqrt{8\log(1.25/\delta)}}{\varepsilon}$, $\varepsilon > 0$, and $0 < \delta < 1$.

Algorithm 1: Skipping Scheme

```
Input: Datasets $\mathcal{D}_v$ and $k$-Lipschitz continuous convex
       functions $f_v: \mathcal{W} \times \mathcal{R}^\kappa \rightarrow \mathbb{R}$, $v \in \mathcal{V}$, in the first
       argument, noise standard deviation sequence
       $(\sigma_1, \ldots, \sigma_{h_{\max}})$, node path sequence
       $(v^{(1)}, \ldots, v^{(h_{\max})})$, learning rate parameter $\zeta$,
       skipping parameter $t_{\text{skip}}$, number of steps $h_{\max}$,
       and communication latency $\chi$
Output: $(\tau_{h_{\max}}, \ell)$
$\tau_0 \leftarrow 0, \quad \ell \leftarrow 0, \quad c \leftarrow 1$
$\mathcal{P} \leftarrow$ Comp. lat. model (gamma or Pareto type II)
for $h \in [h_{\max}]$ do
    $t \sim \mathcal{P}$
    if $t \leq t_{\text{skip}}$ then
        $\eta_h \leftarrow \zeta / \sqrt{c}$
        $\tau_h \leftarrow$
        $\Pi_{\mathcal{W}}\big(\tau_{h-1} - \eta_h\big(\nabla f_{v^{(h)}}(\tau_{h-1}; \mathcal{D}_{v^{(h)}}) + N_h\big)\big)$, where
        $N_h \sim \mathcal{N}(0, \sigma_h^2 I_d)$
        $\ell \leftarrow \ell + \chi + t, \quad c \leftarrow c + 1$
    else
        $\tau_h \leftarrow \tau_{h-1}, \quad \ell \leftarrow \ell + \chi + t_{\text{skip}}$
return $(\tau_{h_{\max}}, \ell)$
```

it reduces to a scheme for which the token always waits. We denote by $p = \Pr[T > t_{\text{skip}}]$ the probability of skipping a node. The formal algorithm is given in Algorithm 1, where the output ℓ denotes its execution latency and $\tau_{h_{\max}}$ the final value of the token after $h_{\max}$ steps.

We use Algorithm 1 in two special cases as outlined below and illustrated in Fig. 1. For both schemes, the noise variance is fixed throughout the algorithm, i.e., $\sigma_h = \sigma$, $\forall h$, and we assume, for simplicity, that $h_{\max}$ is a multiple of n in the rest of the paper.

- First, we consider an update schedule in which the nodes in $\mathcal{V}$ are processed along a logical ring, i.e., the node path sequence of Algorithm 1 is $(v^{(1)}, \ldots, v^{(h_{\max})}) = ((v_1, \ldots, v_n), (v_1, \ldots, v_n), \ldots, (v_1, \ldots, v_n))$. The corresponding scheme is denoted by `Skip-Ring`.
- Second, we consider a *randomized* version of the logical ring, denoted by `Skip-Rand-Ring`. Each round over the ring can be seen as a random walk on the set of nodes, but without replacement. For each round, the random walk procedure is restarted. Hence, the node path sequence becomes $(v^{(1)}, \ldots, v^{(h_{\max})}) = ((v_{\pi_1(1)}, \ldots, v_{\pi_1(n)}), (v_{\pi_2(1)}, \ldots, v_{\pi_2(n)}), \ldots, (v_{\pi_{h_{\max}/n}(1)}, \ldots, v_{\pi_{h_{\max}/n}(n)}))$ where $\pi_1, \ldots, \pi_{h_{\max}/n}$ are independent random permutations over $[n]$.

As a final remark, we mention here that results on the computation and communication latency for the skipping scheme in Algorithm 1 will be presented later in Section VI-A.

IV. Convergence Analysis

Here, we provide a convergence result for the two considered schemes by adapting the classical convergence result of [52, Th. 2] to decentralized learning where nodes are processed according to a Markov chain and for which the (sub)gradient estimate in each step is biased, but *converges to unbiased* exponentially fast, which are the main two new technicalities of the proof.[5] Additionally, the number of token updates is random (depending on the skipping probability), and we need to average over it. Note that, as in [52, Th. 2], f_v, $v \in \mathcal{V}$, is not required to be β-smooth or even k-Lipschitz continuous, as we only need the (sub)gradients to be bounded (which follows from k-Lipschitzness), and also that our result provides a guarantee on the performance of the last update of the token instead of for the average of all token values.

Theorem 1: Under Assumption 1, if the diameter of $\mathcal{W}$ is $d_{\mathcal{W}}$, the expected difference between the minimum value $f(\tau^*; \cdot)$ and that from Algorithm 1 with an arbitrary learning rate parameter $\zeta > 0$ after $h_{\max}$ steps is bounded as

$$\begin{aligned}\mathbb{E}\big[f(\tau_{h_{\max}}; \cdot) - f(\tau^*; \cdot)\big] &\leq \sum_{h=0}^{h_{\max}} \binom{h_{\max}}{h} (1-p)^h p^{h_{\max}-h} e_h \\ &= O\Big(\frac{\log(h_{\max})}{\sqrt{h_{\max}}}\Big),\end{aligned}$$

where $\forall\, h > 0$,

$$\begin{aligned}e_h \triangleq\ & \frac{\Big(d_{\mathcal{W}}^2 + \zeta^2(k^2 + d\sigma^2)\Big)(2 + \log(h+1))}{\zeta\sqrt{h+1}} \\ & + d_{\mathcal{W}} k \sqrt{n}\Big(\frac{1}{h+1}\sum_{i=1}^{h+1} |\lambda_1|^i + \sum_{j=1}^{h} \frac{1}{j(j+1)} \sum_{i=h+1-j}^{h+1} |\lambda_1|^i\Big)\end{aligned}$$

and $e_0 \triangleq d_{\mathcal{W}} k$, $|\lambda_1| = \frac{1-p}{\sqrt{(1+p^2)-2p\cos(\frac{2\pi}{n})}}$ and $0 < p < 1$ for `Skip-Ring`, while $\lambda_1 \triangleq 0$ and $0 \leq p < 1$ for `Skip-Rand-Ring`.

Proof: See Appendices A and E in the supplementary material for the finite and asymptotic results, respectively. ■

Note that the asymptotic convergence rate is the same as that of [52, Th. 2], while being a $\log(h_{\max})$-factor worse compared to [60, Th. 1]. The latter is due to 1) the assumption that σ_h decays to zero with h [60, eq. (16)], and 2) that convergence there is proved for the running average of the token.

Interestingly, the asymptotic behavior of the bound in Theorem 1 is the same for both $\lambda_1 = 0$ and $\lambda_1 > 0$. Hence, a biased (sub)gradient estimate that converges to unbiased exponentially fast does not influence the asymptotic convergence rate. Moreover, in Theorem 1, we do not allow for $p = 0$ in the `Skip-Ring` scheme as in this case the stochastic (sub)gradient is biased, even asymptotically, and hence a different proof technique is required. The asymptotic convergence rate in this special corner case is left open. Note that the proof of [52, Th. 2] cannot be adapted to this scenario as it requires an unbiased stochastic (sub)gradient.

Remark 1: For the uniform random walk scheme considered in [11], the marginal distribution of visited nodes at each step is uniform, as it is with `Skip-Rand-Ring`. Thus, the

[5] There are several previous works that provide convergence results for Markov chain (noisy) stochastic gradient descent, e.g., [60], [61]. However, all of these works require that σ_h decays to zero with h, which means a significantly higher leakage of private data.

proof of Theorem 1 applies to both these schemes with $\lambda_1 \triangleq 0$ and $0 \leq p < 1$.

V. Privacy Analysis

In this section, we present results on the privacy leakage level of the skipping scheme for both updating schedules of the token outlined in Section III-A, i.e., for both a fixed and a randomized logical ring on the set of nodes $\mathcal{V}$. We highlight here that compared to [11], that only considers a constant learning rate and also a different randomized path (and no fixed path), our results apply to a decreasing learning rate of the form $\eta_h = \zeta/\sqrt{h}$ (as specified in Algorithm 1).

The full proof, which can be found in Appendix B, revolves around upper bounding the Rényi divergence between $\mathcal{O}_v(\mathcal{A}(\mathcal{D}))$ and $\mathcal{O}_v(\mathcal{A}(\mathcal{D}'))$, $\mathcal{D} \sim_u \mathcal{D}'$, for any distinct pair of users u, v, using tools (including a composition theorem for Rényi DP (RDP) [62, Proposition 1]) from the framework of privacy amplification by iteration [13]. The resulting bound can be transformed into a bound on NDP using [62, Proposition 3] and further optimized. Allowing for a decreasing learning rate constitutes the main technical contribution of the proof.

Theorem 2: Let $\varepsilon > 0$ and $0 < \delta < 1$. Then, under Assumptions 1 and 2, the `Skip-Ring` scheme on a ring with n nodes and with learning rate parameter $0 < \zeta \leq 2/\beta$ achieves $(\varepsilon_{\text{skip}}, \delta + \delta')$-NDP for all $\delta' \in (0, 1]$ with

$$\varepsilon_{\text{skip}} = \varepsilon \frac{\sqrt{\bar{h} \log(1/\delta)}}{\sqrt{\log(1.25/\delta)}} + \frac{\varepsilon^2 \bar{h}}{4 \log(1.25/\delta)},$$

where $\bar{h} \triangleq \lceil h_{\max}(1-p)/n + \sqrt{3h_{\max}(1-p)/n \log(1/\delta')} \rceil$ and $0 \leq p < 1$ is the probability of skipping a node.

The following theorem characterizes the privacy leakage level $\varepsilon_{\text{skip}}$ of the `Skip-Rand-Ring` scheme.

Theorem 3: Let $\varepsilon > 0$ and $0 < \delta < 1$. Then, under Assumptions 1 and 2, the `Skip-Rand-Ring` scheme on a ring with n nodes and with learning rate parameter $0 < \zeta \leq 2/\beta$ achieves $(\varepsilon_{\text{skip}}, \delta + \delta')$-NDP for all $\delta' \in (0, 1]$ with

$$\varepsilon_{\text{skip}} = \frac{\varepsilon^2 a \alpha}{2 \log(1.25/\delta)} + \frac{\log(1/\delta)}{\alpha - 1},$$

where

$$a \triangleq \frac{1}{n-1} \sum_{r=0}^{\bar{h}-1} \sum_{d=1}^{n-1} \sum_{h=1}^{d} \frac{h \binom{d}{h} p^{d-h} (1-p)^h}{\gamma_{r,h}},$$

$$\gamma_{r,h} \triangleq 4(1 + r \cdot h) \cdot \left(\sqrt{1 + r \cdot h + h} - \sqrt{1 + r \cdot h} \right)^2,$$

$$\bar{h} \triangleq \left\lceil h_{\max}(1-p)/n + \sqrt{3h_{\max}(1-p)/n \log(1/\delta')} \right\rceil,$$

$$\alpha \triangleq \min\left(\frac{\sqrt{2 \log(1/\delta) \log(1.25/\delta)}}{\varepsilon \sqrt{a}} + 1, \frac{1 + \sqrt{\frac{16 \log(1.25/\delta)}{\varepsilon^2} + 1}}{2} \right),$$

and $0 \leq p < 1$ is the probability of skipping a node.[6]

Remark 2: It follows from Theorems 2 and 3 that the asymptotic behavior of the privacy leakage level $\varepsilon_{\text{skip}}$ for both `Skip-Ring` and `Skip-Rand-Ring` is linear in $h_{\max}$, i.e., $\varepsilon_{\text{skip}} = O(h_{\max})$, for $0 \leq p < 1$.

As a final remark, the privacy analysis relies on the exact number of updates performed. Skipping introduces uncertainty on which nodes participated and can be seen as a way to realize *subsampling* [12] on the fly.

VI. Experiments

Here, we first present some results on the computation and communication latency for the skipping scheme in Algorithm 1 that will be used in the numerical results.

Second, we perform a comparison based on the analytical results from Sections IV and V, before turning to training a logistic regression model using the dataset in [53] and a deep neural network for image classification using the MNIST [54] and CIFAR-10 [55] datasets. Finally, we compare with a parallel and a centralized federated learning approach.

A. Computation and Communication Latency

The average total latency of the skipping scheme in Algorithm 1 is given by the following lemma.

Lemma 1: The expected total latency for the skipping scheme in Algorithm 1 is

$$h_{\max} \left(\chi + \int_0^{t_{\text{skip}}} t \, \mathrm{d}\Phi_T(t) + t_{\text{skip}} \big(1 - \Phi_T(t_{\text{skip}})\big) \right),$$

where $\Phi_T(t) \triangleq \Pr[T \leq t]$ and $\Phi_T(t_{\text{skip}}) = 1 - p$.

If the number of hops $h_{\max}$ is large enough, we would expect shorter times between token updates (all other properties being the same) to be beneficial for convergence. In other words, expected time between two consecutive visits to Line 7 in Algorithm 1 should be minimized.

Lemma 2: The value of t_{skip} that minimizes the average time between two consecutive updates of the token is given by the solution of the optimization problem[7]

$$\arg\min_{t_{\text{skip}}} \frac{\chi + \int_0^{t_{\text{skip}}} t \, \mathrm{d}\Phi_T(t) + t_{\text{skip}} \big(1 - \Phi_T(t_{\text{skip}})\big)}{\Phi_T(t_{\text{skip}})}.$$

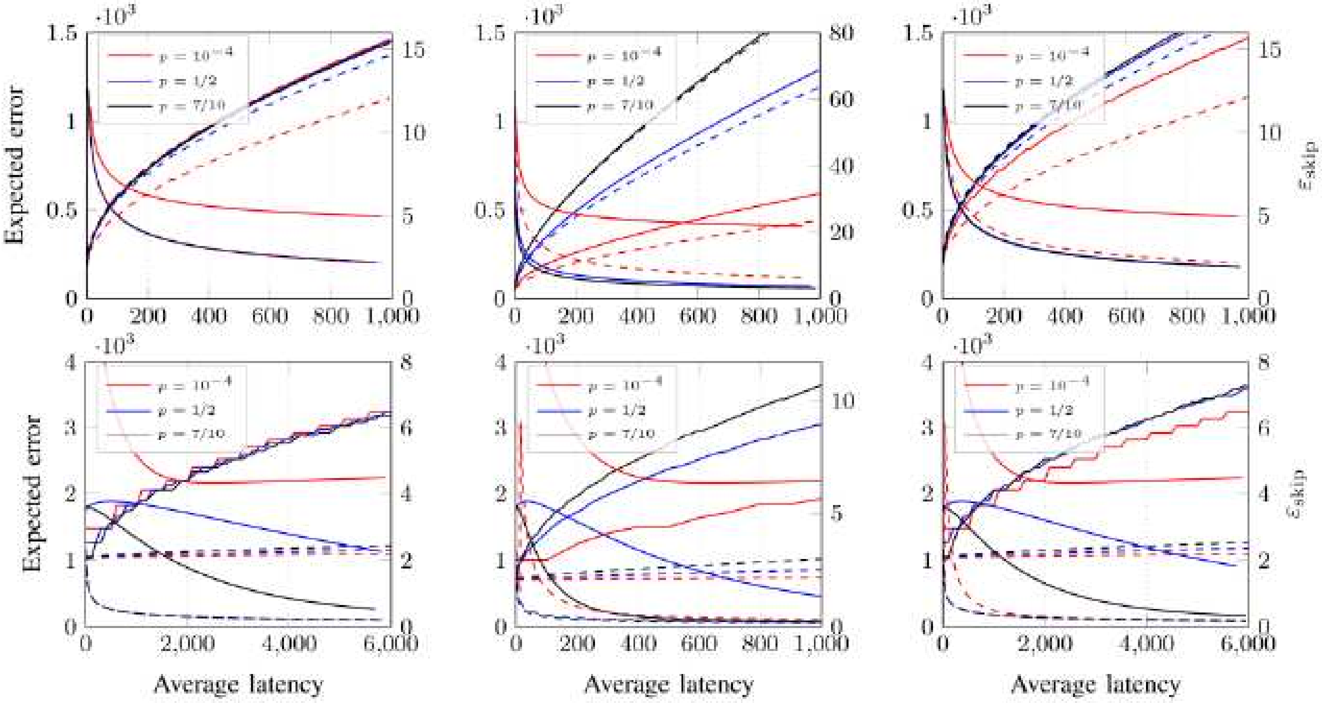
B. Convergence Versus Privacy and Average Latency

We fix $\varepsilon = 1$, $\delta = 10^{-6}$, $\delta' = 10^{-6}$, $d = 8$, $d_{\mathcal{W}} = 10$, $k = 1$, $\zeta = 6/10$, and $\chi = 1/100$. Results are presented for two different values of the number of nodes n, namely for a small number of $n = 10$ nodes and a large number of $n = 500$ nodes.[8] The three characteristics we are interested in are: average latency, expected error bound, and privacy leakage level $\varepsilon_{\text{skip}}$. We first consider $n = 10$ nodes. The top row of Fig. 2 plots expected error bound (left y-axis) and privacy leakage level (right y-axis)

[6] For the uniform random walk scheme considered in [11], a similar result can be derived (see Theorem 4 in Appendix C in the supplementary material). In fact, the only distinction lies in a different definition of the parameter a. However, as shown there, the privacy leakage level $\varepsilon_{\text{skip}}$ is higher compared to the `Skip-Rand-Ring` scheme.

[7] Note that the *optimal* value of t_{skip} can incorporate the probability of link failures and channel noise between nodes by changing the distribution of T.

[8] In [63] and [64], [65], a rather small number of nodes ($n = 25$ and $n = 15$ or 20, respectively) was used in all numerical results, while in [11], [16] a rather large number of nodes ($n = 1000$, 2000, or 4000) was used.



(a) Exponential (mean 1). (b) Gamma (shape 1/4, scale 1). (c) Pareto type II (shape 3, scale 2).

Fig. 2. Expected error bound (decreasing curves; Theorem 1) and privacy leakage level $\varepsilon_{\text{skip}}$ (increasing curves; Theorems 2 and 3) vs average latency (Lemma 1) for $n = 10$ (top row) and $n = 500$ (bottom row). Solid lines are for a fixed ring (Skip-Ring), while dashed lines are for Skip-Rand-Ring.

versus average latency, and the top row of Fig. 3 shows privacy leakage level versus expected error bound, illustrating the inherent trade-off between average latency, expected error bound, and privacy leakage level. The plots are for the three latency models: exponential with mean 1, gamma with shape 1/4 and scale 1, and Pareto type II with shape 3 and scale 2 (as used in [59]). The probability of skipping $p = \Pr[T > t_{\text{skip}}] \in \{10^{-4}, 1/2, 7/10\}$, since $p = 10^{-4}$ and 7/10 are close to the values of p corresponding to the optimal values of t_{skip} given by Lemma 2, respectively 0/0.710/0.737 for the exponential/gamma/Pareto delay models, while $p = 1/2$ is a value in between.[9]

As can be seen from the plots, $p = 10^{-4}$ (virtually, no skipping) gives the worst expected error bound for all considered latency models, for both schemes. This is particularly evident for Skip-Ring with $n = 500$ (second row of plots), where the convergence rate is noticeably slow due to $|\lambda_1| \approx 1 - 10^{-8}$, which is very close to 1. On the other hand, this value of p provides the best privacy leakage level for the same average latency. Hence, there is a trade-off between privacy and accuracy of the algorithm (cf. the top row of plots in Fig. 3), and one needs to choose the skipping probability based on a particular optimization problem.

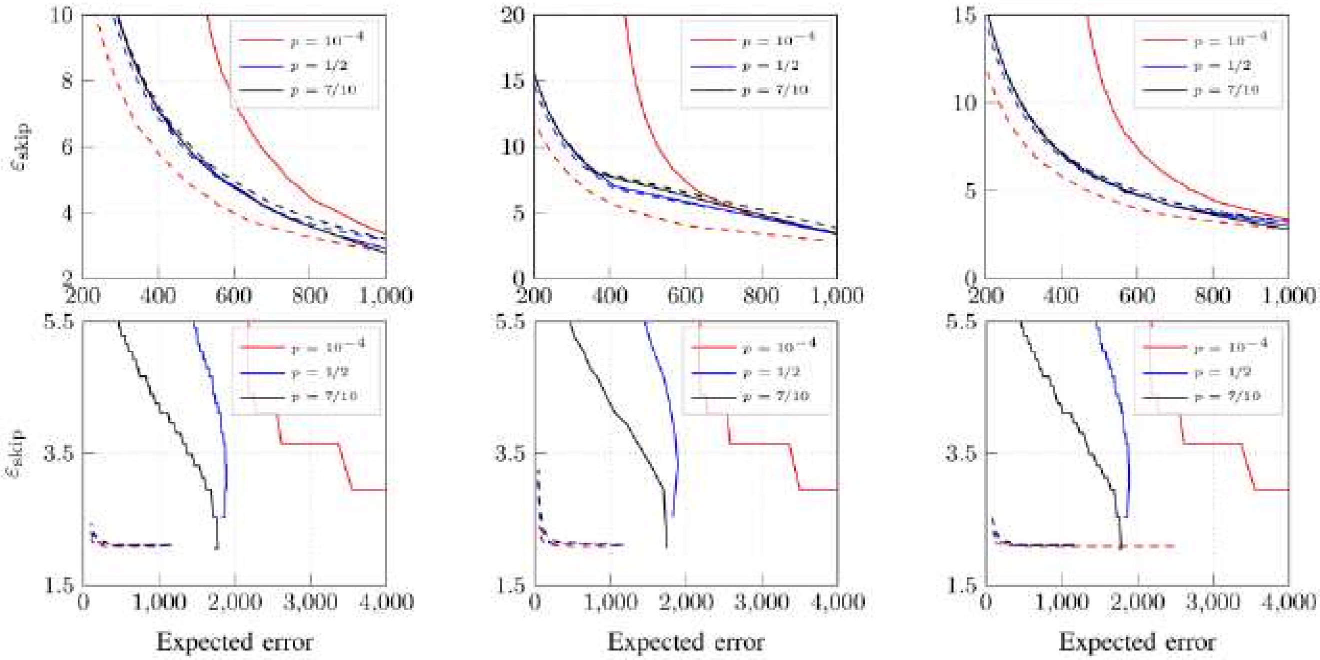
The privacy-versus-error trade-offs look similar for all latency models considered. Skip-Rand-Ring gives better trade-off curves (especially for $p = 10^{-4}$) for smaller values of expected error bound, while the situation changes for higher values of error (i.e., at the initial stages of Algorithm 1's execution). Hence, path randomization improves the trade-off in the long run, but might be harder to realize in a real-world implementation as it would require a full mesh topology.[10]

On the contrary, the Skip-Ring curve for $p = 10^{-4}$ is the worst, which means that skipping helps. Also, there is not much difference between the Skip-Ring curves for $p = 1/2$ and $p = 7/10$ (they are are almost on top of each other and hence difficult to distinguish). On the other hand, Skip-Rand-Ring favors smaller values of p (i.e., larger timeout) at the expense of a higher training latency as shown in the next subsection.

In the bottom rows of Figs. 2 and 3, we show the corresponding results for $n = 500$ nodes. As expected, the relative order of the curves remains for the most part the same as for $n = 10$ nodes (compare with the top rows of the figures). We also observe from Fig. 3 that for a given expected error bound the privacy leakage level $\varepsilon_{\text{skip}}$ is lower with $n = 500$ than with $n = 10$ nodes, i.e., privacy amplification kicks in to a larger extent with a larger number of nodes. Also, the Skip-Rand-Ring scheme shows in general a much bigger privacy advantage compared to the Skip-Ring scheme as the privacy amplification effect is stronger with randomization. Finally, note the more pronounced staircase behavior for the

[9] We have picked $p = 10^{-4}$ instead of $p = 0$ as Theorem 1 requires $p > 0$ in the Skip-Ring scheme.

[10] Strictly speaking a full mesh topology is also required for Skip-Ring, as for a high skipping probability p there could potentially be a need for every single node to be able to communicate with all other nodes, while with no skipping only one output communication channel per node is required. However, as p is constant, and the unavailability is assumed independent from one node to another, a few edges should guarantee that at least one node will answer. The probability that more than l edges would be required is p^l, which quickly becomes small, e.g., for $p = 1/2$ and 10 edges, the probability is less than 10^{-3}.



(a) Exponential (mean 1).

(b) Gamma (shape 1/4, scale 1).

(c) Pareto type II (shape 3, scale 2).

Fig. 3. Privacy leakage level $\varepsilon_{\text{skip}}$ vs expected error bound for $n = 10$ (top row) and $n = 500$ (bottom row). Solid lines are for a fixed ring (Skip-Ring), while dashed lines are for Skip-Rand-Ring.

privacy leakage level. This is due to the factor $(1-p)/n$ inside the ceiling function in the definition of $\tilde{h}$ in Theorems 2 and 3, which also explains why the steps are wider for a larger n.

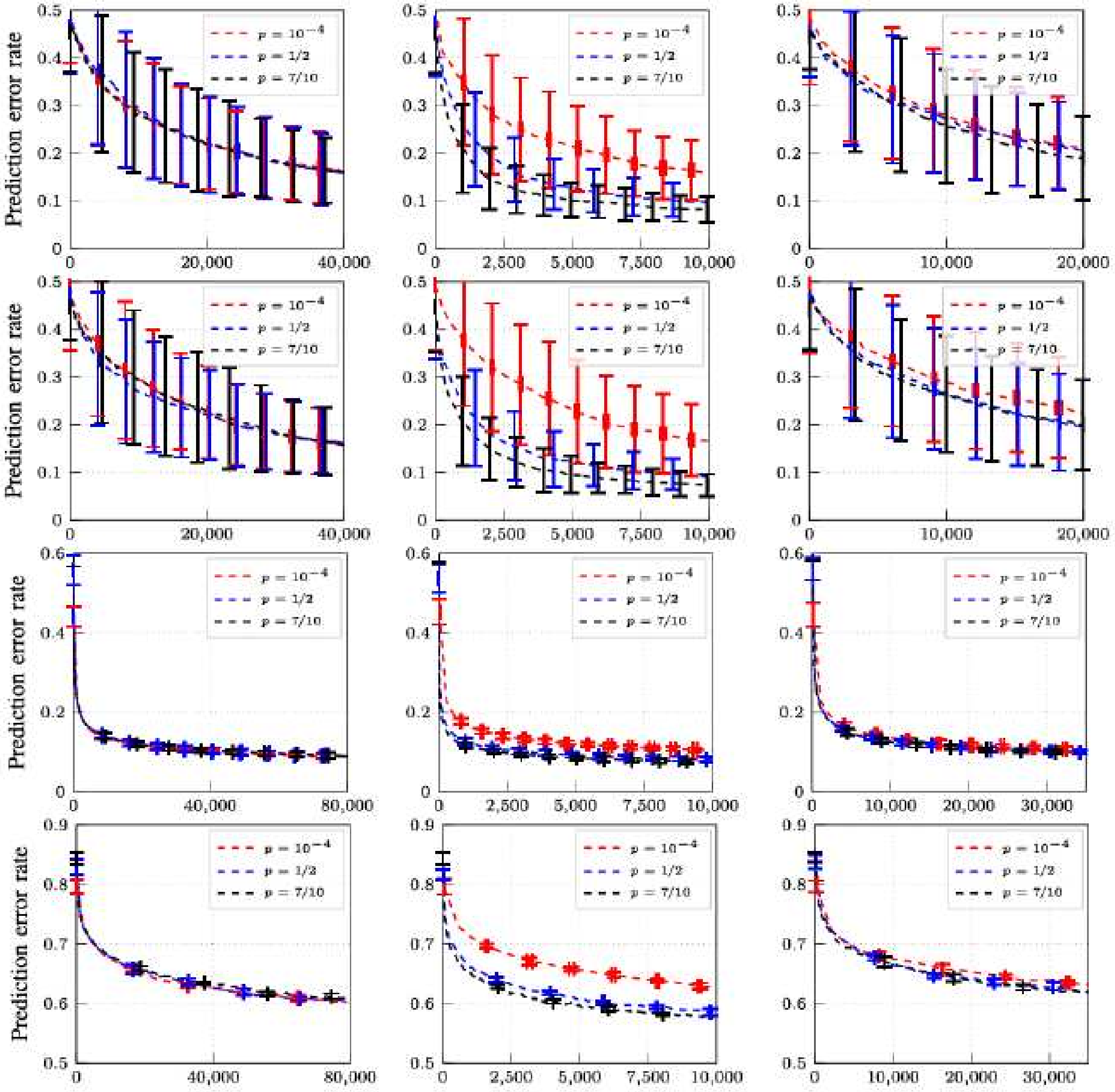
C. Empirical Results

We consider both training a logistic regression model and image classification trained on the MNIST [54] and CIFAR-10 [55] datasets.

1) Logistic Regression: For logistic regression the local loss functions are $f_v(\tau, \mathcal{D}_v) = 1/|\mathcal{D}_v| \sum_{(x,y)\in\mathcal{D}_v} \log(1+\mathrm{e}^{-y\tau x^\top})$, where $x \in \mathbb{R}^d$ ($d_x = d$) and $y \in \{-1, 1\}$ ($d_y = 1$). We use a binarized version of the UCI housing dataset [53], trying to predict binary variable y (whether house price is above a threshold) from other features, x. The features are standardized and we further normalize each data point to have unit ℓ_2-norm so that the loss functions $f_v(\tau; \mathcal{D}_v)$ are 1-Lipschitz continuous (i.e., $k = 1$). The dataset is split uniformly at random into a training set with 80% of the data points and a test set with 20% of the points. Moreover, the training dataset is further randomly split across the n nodes in $\mathcal{V}$. We used the Skip-Rand-Ring scheme (similar results are obtained with the Skip-Ring scheme) with the same parameters as in Section VI-B, but using a mini-batch implementation with batches of size 100 and 8 and with an initial learning rate of $\zeta = 6/10$ and $\zeta = 3/10$ for, respectively, $n = 10$ and $n = 1000$ nodes in order to speed up the learning. The chosen mini-batch size is a compromise between the two corner cases: a mini-batch size of 1 is difficult to parallelize, whereas a large mini-batch size may exceed the nodes' limited parallelization capabilities.

For $n = 10$ nodes, the results of the training are shown in the top plots in Fig. 4, which show the prediction error rate, i.e., the ratio of incorrect predictions on the test set, versus average latency from Lemma 1 for the same skipping probabilities as in the corresponding plots in Figs. 2 and 3. We observe that skipping achieves a clear speed-up compared to no skipping, except for the exponential delay model (as predicted well by Lemma 2, which suggests an optimal $t_{\text{skip}} = +\infty$ for the exponential model). This rhymes well with theoretical expected error bounds (dashed curves of the plots in Fig. 2). As can be seen from the plots of Fig. 3, no skipping in general provides a slightly higher privacy for Skip-Rand-Ring. In the second row of plots in Fig. 4, we show the corresponding results with $n = 1000$ nodes. As expected, the main conclusions remain the same as for $n = 10$. In order to have smooth curves the average of 200 independent runs is presented for both $n = 10$ and $n = 1000$ nodes.

2) Image Classification: We consider both the MNIST and CIFAR-10 datasets. Both datasets are commonly-used benchmarks and are comprised of 10 classes of images; MNIST being comprised of 28×28 pixels grayscale images of handwritten digits from 0 to 9, while CIFAR-10 being comprised of 32×32 pixels color images. The number of training samples is 60000 (6000 for each digit) and 50000 (5000 for each class) for the MNIST and CIFAR-10 datasets, respectively. As for logistic regression in Section VI-C1, the training dataset is further randomly split across a number of nodes n in $\mathcal{V}$. While we used $n = 10$ and $n = 1000$ nodes in Section VI-C1, we use $n = 60$ and $n = 50$ nodes, respectively, for the MNIST and CIFAR-10 datasets. As for logistic regression, we use the Skip-Rand-Ring scheme with the same parameters as in Section VI-B, but with a



(a) Exponential (mean 1). (b) Gamma (shape 1/4, scale 1). (c) Pareto type II (shape 3, scale 2).

Fig. 4. Plots (from top): 1) logistic regression model training with $n = 10$ nodes, showing accuracy (on the test set) vs average latency; 2) logistic regression model training with $n = 1000$ nodes; 3) image classification using the MNIST dataset with $n = 60$ nodes; 4) image classification using the CIFAR-10 dataset with $n = 50$ nodes. First and second row of plots: each curve is an average of 200 independent runs for `Skip-Rand-Ring`, while for the third and fourth row of plots an average of, respectively, 30 and 6 runs is presented (`Skip-Rand-Ring`). Horizontal and vertical error bars illustrate the estimated standard deviation.

smaller initial learning rate of $\zeta = 3/1000$ (MNIST) and $\zeta = 7/10000$ (CIFAR-10), and a batch size of 500, which is half the number of data samples in each node. Moreover, we use a cross-entropy loss function.

The results are depicted in the third and fourth row of plots in Fig. 4, showing the prediction error rate on the test set (comprising 10000 images for both datasets) versus average latency from Lemma 1. For both MNIST (the third row of plots) and CIFAR-10 (the bottom plots), we can make the same observations as for the first and second row of plots (logistic regression); skipping achieves a speed-up compared to no skipping, except for the exponential delay model, as predicted by Lemma 2. Moreover, the order of the curves stays the same across the datasets for a given computational delay model. Note, however, that there is some loss in accuracy due to privacy; the accuracy achieved with the MNIST dataset is close to 90%, while with no privacy requirement an accuracy of around 99% can be reached. For the CIFAR-10 dataset, the accuracy decreases from around 70% to around 42% in the best case. This aligns well with results in the literature, showing a reduction in accuracy due to privacy, which is particularly significant for CIFAR-10, see, e.g., [66]. Compared to the case of logistic regression, the average of only 30 (MNIST) and 6 (CIFAR-10) independent

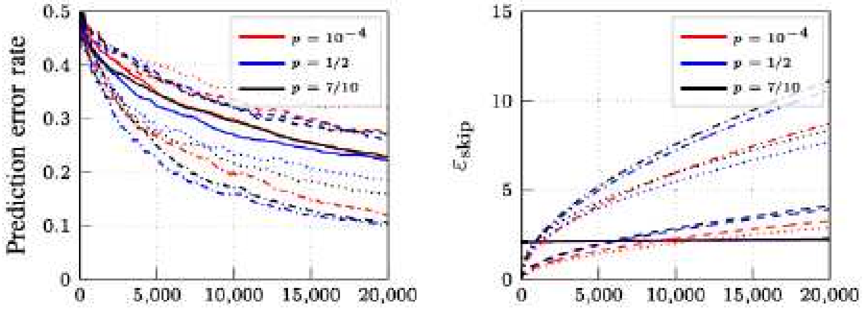


Fig. 5. Comparing Skip-Rand-Ring (solid curves) with Muffliato-SGD ([16, Alg. 3]; dashed and dashdotted curves) and FedL-SGD (dotted curves) for logistic regression model training on the UCI housing dataset with $n = 1000$ nodes for the exponential delay model with mean 1. Left: test set accuracy vs average latency. Right: privacy leakage level vs average latency. Each simulation curve for Muffliato-SGD and FedL-SGD is an average of 100 independent runs, while for Skip-Rand-Ring the average of 200 independent runs is presented. In order to not clutter the plots, no error bars are included.

runs is presented due to the much more complex learning task. The corresponding deep neural networks are detailed in Table I in the supplementary material.

D. Comparisons With a Parallel and a Centralized Federated Learning Approach

For completeness, we also compare our results for logistic regression with a parallel approach using gossip averaging between each step of gradient descent for every node. The most relevant work to compare with is [16]. In [16, Fig. 1(c)], results are presented for logistic regression on the UCI housing dataset [53] of Section VI-C1 using [16, Alg. 3] (Muffliato-SGD). We have replicated the setup of [16, Fig. 1(c)] (using random Erdős-Rényi communication graphs with node degree $\log n$ during gossiping), but with $n = 1000$ nodes, a fixed number of 2 gossip iterations, and $\gamma = 1$ in [16, Alg. 1] (no acceleration) and compare in Fig. 5 Muffliato-SGD (dashed and dashdotted curves) with the Skip-Rand-Ring scheme (solid curves) under the exponential delay model with mean 1. The left plot shows the error prediction rate on the test set, while the right plot shows the (worst-case) privacy leakage level, both as a function of the average latency from Lemma 1. The privacy leakage level for Muffliato-SGD is simulated based on [16, Th. 4] (for two different values of the privacy noise standard deviation; referred to as instances one and two in the next paragraph) and converting to DP using Lemma 11 in Appendix B with $\delta = 10^{-6}$ and with a numerically optimized value of the Rényi divergence parameter α, while for Skip-Rand-Ring we have used the same setup as for the second row of plots in Fig. 4, i.e., Theorem 3 with $\epsilon = 1.0$, $\delta = 10^{-6}$, and $\delta' = 10^{-12}$ (corresponding to a DP noise level of $\sigma_h \approx 10.5976$ used in the actual simulation). We also compare the prediction error rate and the privacy leakage level with those of a centralized federated learning approach (dotted curves), denoted by FedL-SGD in the following.[11] The (worst-case) privacy leakage level for FedL-SGD is computed as for Muffliato-SGD, by converting to DP using Lemma 11 in Appendix B with $\delta = 10^{-6}$ and with a numerically optimized value of the Rényi divergence parameter α. In particular, each time a node u uploads to the central server, $2\alpha/\sigma_h^2$ is added to the overall RDP level of u, and the maximum over all nodes u is the worst-case leakage. We note that implementing gossiping in a latency-efficient manner is not straightforward. In particular, within each iteration of gossiping, each node sends the same information to its neighbors, which can be done through a single broadcast transmission rather than by multiple peer-to-peer transmissions. However, concurrent broadcast transmissions from multiple nodes create interference, which can lead to failed reception of information at the receiver nodes. A simple solution would be through a simple time-division approach in which each node broadcasts sequentially. This entails a communication latency proportional to the number of nodes. A more elaborate approach is random access with broadcast transmission as outlined in [64] or through broadcast-based subgraph sampling as outlined in the very recent paper [65]. For the results in Fig. 5, we have used the random access approach outlined in [64] with an optimized value for the probabilistic random access policy. For FedL-SGD, when computing the training latency, we have assumed 100 independent subchannels for the upload to the central server and a single broadcast transmission to distribute the aggregated gradient back to the nodes. Having a very large number of subchannels would reduce the bandwidth per channel and hence the transmission rate, assuming a fixed overall bandwidth constraint [67], and hence we have used 100 as a compromise (in [67], only 8 or 16 subchannels were used). More details on the latency computation/simulation are given Appendix G in the supplementary material. For Skip-Rand-Ring, we use $\zeta = 3/10$ and a batch size of 8 (as for the second row of plots in Fig. 4), while for Muffliato-SGD and FedL-SGD, we use (as in [16, Fig. 1(c)]) a constant learning rate of $7/10$ and a full batch size of 16 (changing to a batch size of 8 does not noticeably change the accuracy). For a fair comparison, a communication cost of $\chi = 1/100$ is used for all schemes.

From Fig. 5 (left plot), we observe that the Skip-Rand-Ring scheme (solid curves) achieves a low error rate quicker than one of the instances of Muffliato-SGD (dashed curves) and also FedL-SGD

[11] Our simulation of Muffliato-SGD and FedL-SGD is based on https://github.com/totilas/muffliato where gradient clipping is used. For Muffliato-SGD, gradient clipping gives improved accuracy, while for Skip-Rand-Ring we have not observed any noticeable gain with gradient clipping and hence the presented results for Skip-Rand-Ring (as in Fig. 4) are with no clipping.

with virtually no skipping (red dotted curves), while for the second instance of Muffliato-SGD with a lower value of the privacy noise standard deviation (dashdotted curves) and for FedL-SGD with skipping ($p = 1/2$ and $p = 7/10$) we observe the opposite behavior. On the other hand, the overall privacy leakage level grows much slower with the Skip-Rand-Ring scheme (see the right plot). For instance, Muffliato-SGD (second instance; dashdotted curves) achieves an accuracy of 80% quicker than Skip-Rand-Ring (in about 6000 units of time ($p = 1/2$) compared to about 24000 ($p = 10^{-4}$; see first plot in the second row of plots in Fig. 4), but at a much higher privacy leakage level ($\varepsilon_{\text{skip}} = 5.5$ compared to 2.2). Compared to the first instance (dashed curves), however, Skip-Rand-Ring achieves a target accuracy of 80% quicker but at a lower privacy leakage gap (the dashed curves in the right plot lie below the dashdotted curves). FedL-SGD provides a lower privacy leakage level which also grows slower with latency compared to Muffliato-SGD, but on the other hand relies on the assumption of a centralized server. The Skip-Rand-Ring scheme performs favorable compared to FedL-SGD with virtually no skipping, while for $p = 1/2$ and $p = 7/10$ FedL-SGD yields a lower prediction error rate at the expense of a higher privacy leakage compared to Skip-Rand-Ring. In general, smaller values of the privacy noise standard deviation σ_h for FedL-SGD will provide better accuracy, but at the same time increase the privacy leakage level.

VII. Conclusion and Future Work

We have studied a skipping scheme for straggler mitigation in decentralized learning over a logical ring under NDP by extending the framework of privacy amplification by decentralization to include overall training latency—comprising both computation and communication latency. Analytical derivations on both the convergence speed and the DP level were presented, showing a trade-off between overall training latency, accuracy, and user data privacy. The theoretical findings were validated for logistic regression on a real-world dataset and for image classification using the MNIST and CIFAR-10 datasets.

Future work could extend the theoretical analysis in this study to gossip algorithms as examined in [16].

Appendix A
Proof of Theorem 1

A. Notation

Define $[a{:}b] \triangleq \{a, \dots, b\}$ for integers $a \leq b$. Moreover, U^* denotes the conjugate transpose of a matrix U, while U^{-1} denotes its inverse (for a full-rank square matrix U). $\operatorname{diag}(a_1, \dots, a_l)$ denotes an $l \times l$ diagonal matrix with $a_1, \dots, a_l$ along the diagonal.

B. Preliminaries

For the convergence, what matters is only the nodes that actually contributed to the token updates (nonstragglers, i.e., those that reached Line 7 of Algorithm 1). Let $H \in [0 : h_{\max}]$ be the RV denoting the number of nonstragglers when running Algorithm 1, and let the corresponding nodes visited by the token be denoted by $V^{(1)}, V^{(2)}, \dots, V^{(h)}, \dots, V^{(H)}$. If $H = 0$, then all nodes are straggling, no nodes are visited by the token, and Algorithm 1 simply returns $\tau_0 \triangleq 0$ (i.e., $\tau_{h_{\max}} = \tau_0$). Otherwise (i.e., when $H > 0$), according to Algorithm 1, the token updates are (with some abuse of notation)

$$\tau_h \leftarrow \Pi_{\mathcal{W}}\big(\tau_{h-1} - \eta_h\big(\nabla f_{V^{(h)}}\big(\tau_{h-1}; \mathcal{D}_{V^{(h)}}\big) + N_h\big)\big),$$

for all $h \in [H]$. Note also that $\eta_h = \zeta/\sqrt{h}$. In the rest of this subsection, we assume $H > 0$.

For Skip-Rand-Ring, the marginal distribution of a node $V^{(h)}$ is uniform over $\mathcal{V}$ for any h. For Skip-Ring, the sequence of nodes $V^{(1)}, V^{(2)}, \dots$ forms a Markov chain with state transition probability matrix

$$Q = \frac{1-p}{1-p^n}\begin{pmatrix} p^{n-1} & 1 & p & p^2 & \dots & p^{n-2} \\ p^{n-2} & p^{n-1} & 1 & p & \dots & p^{n-3} \\ \vdots & \vdots & \vdots & & \ddots & \vdots \\ 1 & p & p^2 & p^3 & \dots & p^{n-1} \end{pmatrix}, \tag{3}$$

where the entries $Q_{ij} \triangleq \Pr[V^{(h)} = v_j \mid V^{(h-1)} = v_i]$, $1 \leq i, j \leq n$, $h \geq 1$, and, as we show in Lemma 3 below, the marginal distributions of $V^{(h)}$ converge to the uniform distribution exponentially fast when $h \to \infty$.

The uniform distribution of $V^{(h)}$ for Skip-Rand-Ring ensures an unbiased estimate of the real (sub)gradient for any fixed τ, i.e.,

$$\mathbb{E}_{V^{(h)}}\big[\nabla f_{V^{(h)}}(\tau; \mathcal{D}_{V^{(h)}})\big] = \nabla f(\tau; \mathcal{D}),$$

while for Skip-Ring we have that

$$\mathbb{E}_{V^{(h)}}\big[\nabla f_{V^{(h)}}(\tau; \mathcal{D}_{V^{(h)}})\big] \xrightarrow[h\to\infty]{} \nabla f(\tau; \mathcal{D}).$$

Unbiasness of the (sub)gradient estimate at each step is a known condition used to prove convergence of (conventional) stochastic gradient descent. In this appendix, we will show that having *asymptotically* unbiased estimates is sufficient for the convergence of Algorithm 1 too. More precisely, we will adapt a proof from [52, Th. 2] to our scenario.

First, we present some technical results used in the main part of the proof (next subsection).

Lemma 3: For $n \geq 2$, let $\{V^{(h)}\}$, $V^{(h)} \in \mathcal{V}$, $h \geq 1$, be a homogeneous Markov chain with state transition probability matrix (3) with $0 < p < 1$. If we denote by $\pi^{(h)}$ the probability vector of the marginal distribution of $V^{(h)}$ (i.e., $\Pr[V^{(h)} = v_a] = \pi_a^{(h)}$), then $\pi^{(h)} \to \pi^{(\infty)} = (1/n, 1/n, \dots, 1/n)^\top$, as $h \to \infty$, and for all h,

$$\left\|\pi^{(h)} - \pi^{(\infty)}\right\|_1 \leq \sqrt{n}|\lambda_1|^h, \tag{4}$$

where $|\lambda_1| = \frac{1-p}{\sqrt{1+p^2-2p\cos\frac{2\pi}{n}}}$.

Remark 3: For convenience, we also define the value $\lambda_1 \triangleq 0$ for Skip-Rand-Ring (and any $0 \leq p < 1$). With this notation, (4) holds in both cases.

Remark 4: For any probability vector π, it holds that $\left\|\pi - \pi^{(\infty)}\right\|_1 \leq \sqrt{n}$, and, thus, Lemma 3 technically holds also for $h = 0$.

Lemma 4: Let $N \sim \mathcal{N}(0, \sigma^2 I_d)$. Then, $\mathbb{E}\big[\|N\|_2\big] < \sigma\sqrt{d}$ and $\mathbb{E}\big[\|N\|_2^2\big] = d\sigma^2$.

Lemma 5 ([68, Lemma 2]): If the domain $\mathcal{W} \subset \mathbb{R}^d$ is convex and closed, then for any $x, y \in \mathbb{R}^d$, we have $\|x - y\|_2 \geq \|\Pi_{\mathcal{W}}(x) - \Pi_{\mathcal{W}}(y)\|_2$.

Lemma 6: For any $x, y \in \mathbb{R}^d$, $\|x \pm y\|_2^2 = \|x\|_2^2 + \|y\|_2^2 \pm 2x^\top y$.

C. Main Part of the Proof of Theorem 1

We first consider the case of $H \geq 1$. For convenience, define

$$g_h \triangleq \nabla f(\tau_{h-1}; \mathcal{D}),$$
$$\hat{g}_h \triangleq \nabla f_{V^{(h)}}\big(\tau_{h-1}; \mathcal{D}_{V^{(h)}}\big) + N_h$$

as a shorthand notation for $h \in [H]$. With this notation, the token is updated as $\tau_h \leftarrow \Pi_{\mathcal{W}}(\tau_{h-1} - \eta_h \hat{g}_h)$.

If $V^{(h)}$ is uniformly distributed over $\mathcal{V}$, we have that $\mathbb{E}\big[\hat{g}_h\big] = g_h$ for any fixed τ_{h-1}, and in both schemes,

$$\begin{aligned}\mathbb{E}\Big[\big\|\hat{g}_h\big\|_2^2\Big] &\overset{(a)}{=} \mathbb{E}\Big[\big\|\nabla f_{V^{(h)}}(\tau_{h-1}; \mathcal{D}_{V^{(h)}})\big\|_2^2\Big] + \mathbb{E}\Big[\|N_h\|_2^2\Big] \\ &\quad + 2\mathbb{E}\Big[N_h^\top \nabla f_{V^{(h)}}(\tau_{h-1}; \mathcal{D}_{V^{(h)}})\Big] \\ &\overset{(b)}{=} \mathbb{E}\Big[\big\|\nabla f_{V^{(h)}}(\tau_{h-1}; \mathcal{D}_{V^{(h)}})\big\|_2^2\Big] + \mathbb{E}\Big[\|N_h\|_2^2\Big] \\ &\overset{(c)}{\leq} k^2 + d\sigma^2,\end{aligned}$$

where (a) is from Lemma 6, (b) is because N_h is independent of other RVs and has zero mean, and (c) follows from the k-Lipschitz property of f and Lemma 4.

Now, we prove the main statement of Theorem 1. In the proof, if it is not mentioned explicitly, the norm of a vector is the ℓ_2-norm. Also, we assume the same dataset $\mathcal{D}$ everywhere and thus omit it for brevity.

Assume $H \geq 1$ is fixed (i.e., we condition on it). For any $\tau \in \mathcal{W}$, by Lemma 5,

$$\begin{aligned}&\mathbb{E}\Big[\big\|\Pi_{\mathcal{W}}(\tau_{h-1} - \eta_h \hat{g}_h) - \Pi_{\mathcal{W}}(\tau)\big\|^2\Big] \\ &\quad \leq \mathbb{E}\Big[\big\|\tau_{h-1} - \eta_h \hat{g}_h - \tau\big\|^2\Big].\end{aligned}$$

Thus,

$$\begin{aligned}&\mathbb{E}\Big[\|\tau_h - \tau\|^2\Big] \\ &\quad = \mathbb{E}\Big[\big\|\Pi_{\mathcal{W}}(\tau_{h-1} - \eta_h \hat{g}_h) - \Pi_{\mathcal{W}}(\tau)\big\|^2\Big] \\ &\quad \leq \mathbb{E}\Big[\big\|(\tau_{h-1} - \tau) - \eta_h \hat{g}_h\big\|^2\Big] \\ &\quad = \mathbb{E}\Big[\|\tau_{h-1} - \tau\|^2\Big] + \eta_h^2 \mathbb{E}\Big[\big\|\hat{g}_h\big\|^2\Big] - 2\eta_h \mathbb{E}\Big[(\tau_{h-1} - \tau)^\top \hat{g}_h\Big] \\ &\quad \leq \mathbb{E}\Big[\|\tau_{h-1} - \tau\|^2\Big] + \eta_h^2 (k^2 + d\sigma^2) - 2\eta_h \mathbb{E}\Big[(\tau_{h-1} - \tau)^\top \hat{g}_h\Big] \\ &\quad \leq \mathbb{E}\Big[\|\tau_{h-1} - \tau\|^2\Big] - 2\eta_h \mathbb{E}\Big[(\tau_{h-1} - \tau)^\top g_h\Big] \\ &\qquad + \eta_h^2 (k^2 + d\sigma^2) + 2\eta_h d_{\mathcal{W}} k\sqrt{n} |\lambda_1|^h,\end{aligned}$$

where the term $d_{\mathcal{W}} k\sqrt{n} |\lambda_1|^h$ appears because of the difference between the distributions of $\hat{g}_h$ and g_h (cf. Lemma 3 and Remark 3). Then,

$$\begin{aligned}\mathbb{E}\Big[(\tau_{h-1} - \tau)^\top g_h\Big] &\leq \frac{\mathbb{E}\big[\|\tau_{h-1} - \tau\|^2\big]}{2\eta_h} - \frac{\mathbb{E}\big[\|\tau_h - \tau\|^2\big]}{2\eta_h} \\ &\quad + \frac{\eta_h (k^2 + d\sigma^2)}{2} + d_{\mathcal{W}} k\sqrt{n} |\lambda_1|^h.\end{aligned}$$

Let j be an arbitrary element in $[H-1]$. Then, summing up and re-arranging, we get

$$\begin{aligned}&\sum_{h=H-j}^{H} \mathbb{E}\Big[(\tau_{h-1} - \tau)^\top g_h\Big] \\ &\quad \leq \frac{\mathbb{E}\Big[\|\tau_{H-j-1} - \tau\|^2\Big]}{2\eta_{H-j}} \\ &\qquad + \sum_{h=H-j}^{H-1} \frac{\mathbb{E}\big[\|\tau_h - \tau\|^2\big]}{2}\left(\frac{1}{\eta_{h+1}} - \frac{1}{\eta_h}\right) \\ &\qquad + \frac{k^2 + d\sigma^2}{2} \sum_{h=H-j}^{H} \eta_h + d_{\mathcal{W}} k\sqrt{n} \sum_{h=H-j}^{H} |\lambda_1|^h.\end{aligned}$$

Since $\tau_h, \tau \in \mathcal{W}$, we have that $\|\tau_h - \tau\|^2 \leq d_{\mathcal{W}}^2$. We also substitute η_h with $\zeta/\sqrt{h}$, which gives

$$\begin{aligned}&\sum_{h=H-j}^{H} \mathbb{E}\Big[(\tau_{h-1} - \tau)^\top g_h\Big] \\ &\quad \leq \frac{\mathbb{E}\Big[\|\tau_{H-j-1} - \tau\|^2\Big]\sqrt{H-j}}{2\zeta} + \frac{d_{\mathcal{W}}^2}{2\zeta}\left(\sqrt{H} - \sqrt{H-j}\right) \\ &\qquad + \frac{k^2 + d\sigma^2}{2} \sum_{h=H-j}^{H} \frac{\zeta}{\sqrt{h}} + d_{\mathcal{W}} k\sqrt{n} \sum_{h=H-j}^{H} |\lambda_1|^h.\end{aligned}$$

Here, we can upper bound the sum of inverse square roots as

$$\sum_{h=H-j}^{H} \frac{\zeta}{\sqrt{h}} \leq \int_{H-j-1}^{H} \frac{\zeta}{\sqrt{h}} \mathrm{dh} = 2\zeta\left(\sqrt{H} - \sqrt{H-j-1}\right).$$

Next, by convexity of f, we can lower bound $(\tau_{h-1} - \tau)^\top g_h$ by $f(\tau_{h-1}) - f(\tau)$. Hence,

$$\begin{aligned}&\sum_{h=H-j}^{H} \mathbb{E}\big[f(\tau_{h-1}) - f(\tau)\big] \\ &\quad \leq \sum_{h=H-j}^{H} \mathbb{E}\Big[(\tau_{h-1} - \tau)^\top g_h\Big] \\ &\quad \leq \frac{\mathbb{E}\Big[\|\tau_{H-j-1} - \tau\|^2\Big]\sqrt{H-j}}{2\zeta} + d_{\mathcal{W}} k\sqrt{n} \sum_{h=H-j}^{H} |\lambda_1|^h \\ &\qquad + \frac{d_{\mathcal{W}}^2}{2\zeta}\left(\sqrt{H} - \sqrt{H-j}\right) \\ &\qquad + \zeta(k^2 + d\sigma^2)\left(\sqrt{H} - \sqrt{H-j-1}\right) \\ &\quad < \frac{\mathbb{E}\Big[\|\tau_{H-j-1} - \tau\|^2\Big]\sqrt{H-j}}{2\zeta} + d_{\mathcal{W}} k\sqrt{n} \sum_{h=H-j}^{H} |\lambda_1|^h \\ &\qquad + \left(\frac{d_{\mathcal{W}}^2}{2\zeta} + \zeta(k^2 + d\sigma^2)\right)\left(\sqrt{H} - \sqrt{H-j-1}\right) \\ &\quad = \frac{\mathbb{E}\Big[\|\tau_{H-j-1} - \tau\|^2\Big]\sqrt{H-j}}{2\zeta} + d_{\mathcal{W}} k\sqrt{n} \sum_{h=H-j}^{H} |\lambda_1|^h\end{aligned}$$

$$
\begin{aligned}
&+ \left(\frac{d_{\mathcal{W}}^2}{2\zeta} + \zeta(k^2 + d\sigma^2)\right)\frac{j+1}{\sqrt{H} + \sqrt{H-j-1}} \\
&< \frac{\mathbb{E}\left[\|\tau_{H-j-1} - \tau\|^2\right]\sqrt{H-j}}{2\zeta} + d_{\mathcal{W}}k\sqrt{n}\sum_{h=H-j}^{H}|\lambda_1|^h \\
&+ \left(\frac{d_{\mathcal{W}}^2}{2\zeta} + \zeta(k^2 + d\sigma^2)\right)\frac{j+1}{\sqrt{H}}. \qquad (5)
\end{aligned}
$$

By setting $\tau = \tau_{H-j-1}$ in (5), we get

$$
\begin{aligned}
&\sum_{h=H-j}^{H}\mathbb{E}\big[f(\tau_{h-1}) - f(\tau_{H-j-1})\big] \\
&\quad \le \left(\frac{d_{\mathcal{W}}^2}{2\zeta} + \zeta(k^2 + d\sigma^2)\right)\frac{j+1}{\sqrt{H}} + d_{\mathcal{W}}k\sqrt{n}\sum_{h=H-j}^{H}|\lambda_1|^h.
\end{aligned}
$$

Next, as a shorthand, let S_j denote the average of the following $j+1$ iterates: $S_j = \frac{1}{j+1}\sum_{h=H-j}^{H} f(\tau_{h-1})$. Then,

$$
\begin{aligned}
&(j+1)\mathbb{E}[S_j] - (j+1)\mathbb{E}\big[f(\tau_{H-j-1})\big] \\
&= \sum_{h=H-j}^{H}\mathbb{E}\big[f(\tau_{h-1}) - f(\tau_{H-j-1})\big] \\
&\le \left(\frac{d_{\mathcal{W}}^2}{2\zeta} + \zeta(k^2 + d\sigma^2)\right)\frac{j+1}{\sqrt{H}} + d_{\mathcal{W}}k\sqrt{n}\sum_{h=H-j}^{H}|\lambda_1|^h.
\end{aligned}
$$

Hence,

$$
\begin{aligned}
-\mathbb{E}\big[f(\tau_{H-j-1})\big] &\le -\mathbb{E}[S_j] + \left(\frac{d_{\mathcal{W}}^2}{2\zeta} + \zeta(k^2 + d\sigma^2)\right)\frac{1}{\sqrt{H}} \\
&\quad + \frac{d_{\mathcal{W}}k\sqrt{n}}{j+1}\sum_{h=H-j}^{H}|\lambda_1|^h.
\end{aligned}
$$

Using this, we have

$$
\begin{aligned}
\mathbb{E}[S_{j-1}] &= \frac{(j+1)\mathbb{E}[S_j] - \mathbb{E}\big[f(\tau_{H-j-1})\big]}{j} \\
&\le \mathbb{E}[S_j] + \left(\frac{d_{\mathcal{W}}^2}{2\zeta} + \zeta(k^2 + d\sigma^2)\right)\frac{1}{j\sqrt{H}} \\
&\quad + \frac{d_{\mathcal{W}}k\sqrt{n}}{j(j+1)}\sum_{h=H-j}^{H}|\lambda_1|^h.
\end{aligned}
$$

In the following, to simplify notation, define

$$
a_j \triangleq \left(\frac{d_{\mathcal{W}}^2}{2\zeta} + \zeta(k^2 + d\sigma^2)\right)\frac{1}{j\sqrt{H}} + \frac{d_{\mathcal{W}}k\sqrt{n}}{j(j+1)}\sum_{h=H-j}^{H}|\lambda_1|^h
$$

as a shorthand. Then,

$$
\begin{aligned}
\mathbb{E}\big[f(\tau_{H-1})\big] &= \mathbb{E}[S_0] \le \mathbb{E}[S_1] + a_1 \le \mathbb{E}[S_2] + a_1 + a_2 \\
&\le \cdots \le \mathbb{E}\big[S_{H-1}\big] + \sum_{j=1}^{H-1} a_j.
\end{aligned}
$$

Next, we bound a part of the sum on the right hand side as

$$
\begin{aligned}
&\sum_{j=1}^{H-1}\left(\frac{d_{\mathcal{W}}^2}{2\zeta} + \zeta\left(k^2 + d\sigma^2\right)\right)\frac{1}{j\sqrt{H}} \\
&\le \sum_{j=1}^{H-1}\left(\frac{d_{\mathcal{W}}^2}{\zeta} + \zeta\left(k^2 + d\sigma^2\right)\right)\frac{1}{j\sqrt{H}} \\
&\le \frac{d_{\mathcal{W}}^2 + \zeta^2\left(k^2 + d\sigma^2\right)}{\zeta\sqrt{H}}(1 + \log H)
\end{aligned}
$$

and obtain

$$
\begin{aligned}
\mathbb{E}\big[f(\tau_{H-1})\big] &\le \mathbb{E}\big[S_{H-1}\big] + \frac{d_{\mathcal{W}}^2 + \zeta^2(k^2 + d\sigma^2)}{\zeta\sqrt{H}}(1 + \log H) \\
&\quad + \sum_{j=1}^{H-1}\frac{d_{\mathcal{W}}k\sqrt{n}}{j(j+1)}\sum_{h=H-j}^{H}|\lambda_1|^h. \qquad (6)
\end{aligned}
$$

Now, recall (5). Set there $j = H-1$ (i.e., $H-j = 1$), $\tau = \tau^*$, and bound all norms by $d_{\mathcal{W}}^2$, which results in

$$
\begin{aligned}
&\sum_{h=1}^{H}\mathbb{E}\big[f(\tau_{h-1}) - f(\tau^*)\big] \\
&\le \frac{d_{\mathcal{W}}^2}{2\zeta} + d_{\mathcal{W}}k\sqrt{n}\sum_{h=1}^{H}|\lambda_1|^h + \left(\frac{d_{\mathcal{W}}^2}{2\zeta} + \zeta(k^2 + d\sigma^2)\right)\sqrt{H} \\
&\le \left(\frac{d_{\mathcal{W}}^2}{\zeta} + \zeta(k^2 + d\sigma^2)\right)\sqrt{H} + d_{\mathcal{W}}k\sqrt{n}\sum_{h=1}^{H}|\lambda_1|^h.
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
\mathbb{E}\big[S_{H-1}\big] - f(\tau^*) &= \mathbb{E}\left[\frac{1}{H}\sum_{h=1}^{H}\big(f(\tau_{h-1}) - f(\tau^*)\big)\right] \\
&\le \frac{d_{\mathcal{W}}^2 + \zeta^2(k^2 + d\sigma^2)}{\zeta\sqrt{H}} + \frac{d_{\mathcal{W}}k\sqrt{n}}{H}\sum_{h=1}^{H}|\lambda_1|^h. \qquad (7)
\end{aligned}
$$

Finally, by combining (6) and (7), we obtain

$$
\begin{aligned}
&\mathbb{E}\big[f(\tau_{H-1}) - f(\tau^*)\big] \\
&\le \frac{\big(d_{\mathcal{W}}^2 + \zeta^2(k^2 + d\sigma^2)\big)(2 + \log H)}{\zeta\sqrt{H}} \\
&\quad + d_{\mathcal{W}}k\sqrt{n}\left(\frac{1}{H}\sum_{h=1}^{H}|\lambda_1|^h + \sum_{j=1}^{H-1}\frac{1}{j(j+1)}\sum_{h=H-j}^{H}|\lambda_1|^h\right).
\end{aligned}
$$

Then,

$$
\begin{aligned}
&\mathbb{E}\big[f(\tau_H) - f(\tau^*)\big] \\
&\le \frac{\left(d_{\mathcal{W}}^2 + \zeta^2(k^2 + d\sigma^2)\right)(2 + \log(H+1))}{\zeta\sqrt{H+1}} \\
&\quad + d_{\mathcal{W}}k\sqrt{n}\left(\frac{1}{H+1}\sum_{h=1}^{H+1}|\lambda_1|^h + \sum_{j=1}^{H}\frac{1}{j(j+1)}\sum_{h=H-j+1}^{H+1}|\lambda_1|^h\right).
\end{aligned}
$$

The corner case of $H = 0$ (and thus, $\tau_{h_{\max}} = \tau_0 = 0$) can be bounded as $|f(0) - f(\tau^*)| \le k\|0 - \tau^*\| \le kd_{\mathcal{W}}$.

As a final step, we need to take expectation conditioned on the distribution of H, which is binomial with $h_{\max}$ independent trials and success probability $1-p$, i.e.,

$$
\Pr[H = h] = \binom{h_{\max}}{h}(1-p)^h p^{h_{\max}-h},
$$

which concludes the proof.

Appendix B
Proof of Theorems 2 and 3

The main tool of the proofs is the concept of privacy amplification by iteration [13], and Theorem 22 therein. The setting in [13] is *projected noisy stochastic gradient decent*, in which noise is added for every gradient update step. The main technical tool is Rényi divergence and the proof evolves around upper bounding it for a single view of a node. In particular, based on Lemma 8, for any distinct pair of users u, v, we can derive an upper bound on the Rényi divergence between the views of user v when the token visits for the $(r+1)$-th time, excluding received and sent messages observed up to and including the r-th visit, for two neighboring datasets of user u (Lemma 12). By maximizing this upper bound over all pairs of distinct users u, v and by using a composition theorem for RDP [62, Proposition 1] (Lemma 9), we can derive an upper bound on the RDP level of Algorithm 1, which can be transformed into an upper bound on the DP level using [62, Proposition 3] (Lemma 11). In order to get the best (lowest) upper bound, the Rényi divergence parameter α can be optimized. Finally, since the number of visits to a node is not a constant, but instead follows a binomial distribution, a standard Chernoff bound in combination with Lemma 10 can be used to derive the final result.

We start by defining Rényi divergence and RDP and then state some important results from the privacy amplification by iteration literature. In particular, definitions and results from [13], [62].

A. Important Results From [13], [62]

We start by stating and adapting some important definitions and results from [13], [62]. Central to the arguments in [13] is the concept of Rényi divergence and shifted Rényi divergence.

Definition 4 (Rényi Divergence): For two probability distributions μ and ν defined over the same set $\mathcal{Z}$, the Rényi divergence of positive order $\alpha \neq 1$ between μ and ν is

$$\mathscr{D}_\alpha(\mu \,\|\, \nu) \triangleq \frac{1}{\alpha - 1} \log \int_{z \in \mathcal{Z}} \left(\frac{\mu(z)}{\nu(z)} \right)^\alpha \nu(z) \text{ dz}.$$

Definition 5 (Shifted Rényi Divergence [13, Definition 8]): For two probability distributions μ and ν defined over the same complete normed vector space $(\mathcal{Z}, \|\cdot\|)$, the u-shifted Rényi divergence, for $u \geq 0$, of order $\alpha > 1$ between μ and ν is

$$\mathscr{D}_\alpha^{(u)}(\mu \,\|\, \nu) \triangleq \inf_{\mu' :\, d_{\mathrm{W}_\infty}(\mu, \mu') \leq u} \mathscr{D}_\alpha\big(\mu' \,\|\, \nu\big),$$

where $d_{\mathrm{W}_\infty}(\cdot, \cdot)$ denotes the ∞-Wasserstein distance [13, Definition 6] between two distributions on $(\mathcal{Z}, \|\cdot\|)$.

Lemma 7 (Weak Convexity Rényi Divergence [13, Lemma 25]): Let $\mu_1, \dots, \mu_n$ and $\nu_1, \dots, \nu_n$ be probability distributions defined on a complete normed vector space $(\mathcal{Z}, \|\cdot\|)$ such that $\forall\, i \in [n]$, $\mathscr{D}_\alpha(\mu_i \,\|\, \nu_i) \leq {}^{b}/_{(\alpha-1)}$ for some $b \in (0, 1]$ where $\alpha > 1$. Let ρ be a probability distribution over $[n]$ and denote by μ_ρ the probability distribution over $\mathcal{Z}$ obtained by sampling i from ρ and then outputting a random sample from μ_i (respectively, ν_i). Then

$$\mathscr{D}_\alpha\big(\mu_\rho \,\big\|\, \nu_\rho\big) \leq (1 + b) \cdot \mathbb{E}_{i \sim \rho}\, \mathscr{D}_\alpha(\mu_i \,\|\, \nu_i).$$

Definition 6 ([13, Definition 10]): For a distribution ζ over $(\mathcal{Z}, \|\cdot\|)$ and any $a \geq 0$, the *magnitude of noise* is the largest Rényi divergence of positive order $\alpha \neq 1$ between ζ and the same distribution ζ shifted by a vector of length at most a, i.e.,

$$\mathscr{R}_\alpha(\zeta, a) \triangleq \sup_{z :\, \|z\| \leq a} \mathscr{D}_\alpha(\zeta \star z \,\|\, \zeta).$$

Remark 5: Consider the standard Gaussian distribution over $\mathbb{R}^d$ with variance σ^2, denoted by $\mathcal{N}(0, \sigma^2 I_d)$. Then, it is known that $\forall\, z \in \mathbb{R}^d, \sigma > 0$ (see, e.g., [69, Ex. 3]), we have

$$\mathscr{D}_\alpha\Big(\mathcal{N}\Big(x, \sigma^2 I_d\Big) \,\Big\|\, \mathcal{N}\Big(0, \sigma^2 I_d\Big)\Big) = \alpha \frac{\|x\|^2}{2\sigma^2},$$
$$\mathscr{R}_\alpha\Big(\mathcal{N}\Big(0, \sigma^2 I_d\Big), a\Big) = \alpha \frac{a^2}{2\sigma^2}.$$

Definition 7 (Contractive Noisy Iteration (CNI) [13, Definition 19]): Given an initial random state $Z_0 \in \mathcal{Z}$, a sequence of contractive maps $\{\psi_h\}_{h=1}^m$, and a sequence of noise distributions $\{\zeta_h\}_{h=1}^m$, the *contractive noisy iteration* after m steps, denoted by $\mathrm{CNI}_m(Z_0, \{\psi_h\}, \{\zeta_h\})$, is defined by the following update process: $Z_h \triangleq \psi_h(Z_{h-1}) + N_h$, where $N_h \sim \zeta_h$, $h \in [m]$.

The following lemma is taken from [13, Th. 22].

Lemma 8 ([13, Th. 22]): Let Z_m and Z'_m represent the outputs of $\mathrm{CNI}_m(Z_0, \{\psi_h\}, \{\zeta_h\})$ and $\mathrm{CNI}_m(Z_0, \{\psi'_h\}, \{\zeta_h\})$, respectively. Define $s_h \triangleq \sup_z \big\|\psi_h(z) - \psi'_h(z)\big\|$, $\{a_h\}_{h=1}^m$ a sequence of nonnegative reals, and $u_h \triangleq \sum_{i=1}^h (s_i - a_i)$. If $u_h \geq 0$, $\forall\, h \in [m]$, then $\mathscr{D}_\alpha^{(u_m)}\big(Z_m \,\big\|\, Z'_m\big) \leq \sum_{h \in [m]} \mathscr{R}_\alpha(\zeta_h, a_h)$.

Now, we review some results from RDP [62].

Definition 8 ((α, ε)-RDP): For any positive $\alpha \neq 1$ and $\varepsilon \geq 0$, a (randomized) protocol $\mathcal{A}$ is said to satisfy (α, ε)-RDP, if for all neighboring datasets $\mathcal{D}, \mathcal{D}'$ and for all $\mathcal{S}$ in the output space Ω, we have $\mathscr{D}_\alpha\big(\mathcal{A}(\mathcal{D}) \in \mathcal{S} \,\big\|\, \mathcal{A}(\mathcal{D}') \in \mathcal{S}\big) \leq \varepsilon$.

Next, we state the composition theorem for RDP.

Lemma 9 ([62, Proposition 1]): Let $r \in \mathbb{N}$. If $\{\mathcal{A}_l\}_{l=1}^r$ are protocols satisfying, respectively, (α, ε_1)-RDP, …, (α, ε_r)-RDP, then their composition defined as $(\mathcal{A}_1, \dots, \mathcal{A}_r)$ satisfies $(\alpha, \sum_{i=1}^r \varepsilon_i)$-RDP.

The DP (RDP) level with a random number of entries in the composition can be bounded as follows.

Lemma 10: Let R denote a RV with range $\{1, 2, \dots\}$ that satisfies $\Pr(R > r) \leq \delta'$. If $\{\mathcal{A}_l\}_{l=1}^R$ are protocols satisfying, respectively, $(\varepsilon_1, \delta_1)$-DP, …, $(\varepsilon_R, \delta_R)$-DP, then their composition defined as $(\mathcal{A}_1, \dots, \mathcal{A}_R)$ satisfies $(\varepsilon_c, \delta_c + \delta')$-DP, where $(\varepsilon_c, \delta_c)$ is the DP guarantee under r-fold composition for DP.

In particular, if R is a binomial RV (i.e., a sum of independent Bernoulli RVs), we can use the standard Chernoff bound to upper bound $\Pr(R > r)$.

A relation between (α, ε)-RDP and (ε, δ)-DP can be stated as follows.

Lemma 11: ([62, Proposition 3]): If $\mathcal{A}$ satisfies (α, ε)-RDP for $\alpha > 1$, then for all $\delta \in (0, 1)$, it also satisfies $(\varepsilon + \frac{\log(1/\delta)}{\alpha - 1}, \delta)$-DP.

B. Adapting to Algorithm 1

For notational convenience, let $\mathcal{O}_v^{(r)}(\mathcal{A}(\mathcal{D}))$ be the view of user v when the token visits for the r-th time, excluding sent/received messages observed up to and including the $(r-1)$-th visit.

The following lemma is analogous to [13, Th. 23], but tailored to our setting with a decreasing learning rate.

Lemma 12: Let $\mathcal{W} \subseteq \mathbb{R}^d$ be a convex set and let $f_v: \mathcal{W} \times \mathcal{R}^\kappa \to \mathbb{R}$, $v \in \mathcal{V}$, be k-Lipschitz continuous and β-smooth convex functions in their first argument. Let $(v_1^{(r+1)}, \dots, v_{l^{(r+1)}}^{(r+1)})$ denote the sequence of nodes visited in between the r-th and $(r+1)$-th visit to node v in Algorithm 1. Then, for Algorithm 1 with learning rate parameter $0 < \zeta \leq {}^2/_\beta$ and constant noise $\sigma_h = \sigma$, and any distinct pair of users $u, v \in \mathcal{V}$,

$$\mathscr{D}_\alpha\Big(\mathcal{O}_v^{(r+1)}(\mathcal{A}(\mathcal{D}))\,\Big\|\,\mathcal{O}_v^{(r+1)}(\mathcal{A}(\mathcal{D}'))\Big)$$
$$\begin{cases} \leq \frac{2\alpha k^2}{\sigma^2} & \text{if } \xi_{u,v}^{(r+1)} = 1, \\ \leq \frac{\alpha k^2 \xi_{u,v}^{(r+1)}}{2\left(1+\sum_{i=1}^r \xi_{u,v}^{(i)}\right)\cdot\left(\sqrt{1+\sum_{i=1}^r \xi_{u,v}^{(i)} + \xi_{u,v}^{(r+1)}} - \sqrt{1+\sum_{i=1}^r \xi_{u,v}^{(i)}}\right)^2 \sigma^2} & \\ & \text{if } 1 < \xi_{u,v}^{(r+1)} < \infty, \\ = 0 & \text{otherwise,} \end{cases}$$

for every $\alpha > 1$, where $\mathcal{D} \sim_u \mathcal{D}'$, $\xi_{u,v}^{(r+1)} \triangleq l^{(r+1)} - c^{(r+1)} + 1$ and $c^{(r+1)} \in [l^{(r+1)}]$ is the index of $v_i^{(r+1)} = u$ for $u \in \{v_1^{(r+1)}, \dots, v_{l^{(r+1)}}^{(r+1)}\}$, i.e., $u = v_{c^{(r+1)}}^{(r+1)}$. Otherwise, if $u \notin \{v_1^{(r+1)}, \dots, v_{l^{(r+1)}}^{(r+1)}\}$, then $\xi_{u,v}^{(r+1)} \triangleq \infty$.

For simplicity of notation, we omit the superscript $(r+1)$ from l, c, and $v_1, \dots, v_l$ in the following.

Proof: Consider the case when $u \in \{v_1, \dots, v_l\}$. Otherwise, $\mathcal{O}_v^{(r+1)}(\mathcal{A}(\mathcal{D})) = \mathcal{O}_v^{(r+1)}(\mathcal{A}(\mathcal{D}'))$, and it follows directly that $\mathscr{D}_\alpha\Big(\mathcal{O}_v^{(r+1)}(\mathcal{A}(\mathcal{D}))\,\Big\|\,\mathcal{O}_v^{(r+1)}(\mathcal{A}(\mathcal{D}'))\Big) = 0$.

By assumption, the learning rate η_{h_i} is upper-bounded by ${}^2/_\beta$, and hence the update rule $g_r^{(v)}(\tau; \mathtt{state}_v(h))$ in (2) for Algorithm 1 constitutes a CNI (see [13, Proposition 18]). Consider now the CNI from Definition 7 with $\psi_i(\tau) = \prod_{\mathcal{W}}(\tau - \eta_{h_i}\nabla f_{v_i}(\tau, \mathcal{D}_{v_i})) = \prod_{\mathcal{W}}(\tau) - \eta_{h_i}\nabla f_{v_i}(\prod_{\mathcal{W}}(\tau), \mathcal{D}_{v_i})$ and with $\psi_h'(\tau) = \prod_{\mathcal{W}}(\tau - \eta_{h_i}\nabla f_{v_i}(\tau, \mathcal{D}_{v_i}')) = \prod_{\mathcal{W}}(\tau) - \eta_{h_i}\nabla f_{v_i}(\prod_{\mathcal{W}}(\tau), \mathcal{D}_{v_i}')$, corresponding to $g_r^{(v)}(\tau; \mathtt{state}_v(h))$ in (2). It follows that

$$\begin{aligned} &\sup_\tau \left\| \psi_i(\tau) - \psi_i'(\tau) \right\|_2 \\ &\quad = \sup_\tau \left\| \eta_{h_i}\nabla f_{v_i}\left(\prod_{\mathcal{W}}(\tau), \mathcal{D}_{v_i}\right) - \eta_{h_i}\nabla f_{v_i}\left(\prod_{\mathcal{W}}(\tau), \mathcal{D}'_{v_i}\right) \right\|_2 \\ &\quad = \begin{cases} 0 & \text{if } i \neq c, \\ \leq 2\eta_{h_c} k & \text{otherwise,} \end{cases} \end{aligned}$$

since by assumption f_{v_i} is k-Lipschitz continuous.

Now apply Lemma 8 with $a_i = 0$, $\forall\, i \in [c-1]$, and $a_i = {}^{2\eta_{h_i}k}/_{\varrho_{u,v}^{(r+1)}}$, $\forall\, i \in [c{:}l]$, where

$$\varrho_{u,v}^{(r+1)} \triangleq \frac{\sum_{i\in[c:l]} \eta_{h_i}}{\eta_{h_c}} = \frac{\sum_{i\in[c:l]} \frac{1}{\sqrt{h_i}}}{\frac{1}{\sqrt{h_c}}}. \tag{8}$$

Clearly, $z_i = s_i - a_i \geq 0$, $\forall\, i \in [l]$, and $z_l = 0$. Hence, using Remark 5,

$$\begin{aligned} &\mathscr{D}_\alpha\Big(\mathcal{O}_v^{(r+1)}(\mathcal{A}(\mathcal{D}))\,\Big\|\,\mathcal{O}_v^{(r+1)}(\mathcal{A}(\mathcal{D}'))\Big) \\ &\quad \leq \alpha \sum_{i\in[c:l]} \frac{4\eta_{h_i}^2 k^2}{2\left(\varrho_{u,v}^{(r+1)}\right)^2 \eta_{h_i}^2 \sigma^2} = \alpha \sum_{i\in[c:l]} \frac{2k^2}{\left(\varrho_{u,v}^{(r+1)}\right)^2 \sigma^2} \\ &\quad = \frac{2\alpha|[c:l]|k^2}{\left(\varrho_{u,v}^{(r+1)}\right)^2 \sigma^2} = \frac{2\alpha\xi_{u,v}^{(r+1)} k^2}{\left(\varrho_{u,v}^{(r+1)}\right)^2 \sigma^2}. \end{aligned} \tag{9}$$

Now, if $c = l$, i.e., $u = v_l$ and $\xi_{u,v}^{(r+1)} = 1$, then from (8) it follows that $\varrho_{u,v}^{(r+1)} = 1$ and therefore

$$\mathscr{D}_\alpha\Big(\mathcal{O}_v^{(r+1)}(\mathcal{A}(\mathcal{D}))\,\Big\|\,\mathcal{O}_v^{(r+1)}(\mathcal{A}(\mathcal{D}'))\Big) \leq \frac{2\alpha k^2}{\sigma^2}.$$

Otherwise, i.e., when $l > c$ and $1 < \xi_{u,v}^{(r+1)} < \infty$,

$$\begin{aligned} \varrho_{u,v}^{(r+1)} &= \frac{\sum_{i\in[c:l]} \frac{1}{\sqrt{h_i}}}{\frac{1}{\sqrt{h_c}}} \\ &\overset{(a)}{>} 2\sqrt{h_c}\left(\sqrt{h_c + \xi_{u,v}^{(r+1)} - 1 + 1} - \sqrt{h_c}\right) \\ &\overset{(b)}{\geq} 2\sqrt{1 + \sum_{i=1}^r \xi_{u,v}^{(i)}} \cdot \left(\sqrt{1 + \sum_{i=1}^r \xi_{u,v}^{(i)} + \xi_{u,v}^{(r+1)}} \right. \\ &\qquad\qquad \left. - \sqrt{1 + \sum_{i=1}^r \xi_{u,v}^{(i)}}\right), \end{aligned} \tag{10}$$

where (a) follows by taking the anti-derivative of ${}^1/_{\sqrt{h_i}}$ and the fact that the learning rate is only updated when visiting a node, i.e., $h_l = h_{l-1} + 1 = h_{l-2} + 2 = \cdots = h_c + l - c$, and (b) follows by lower-bounding h_c by $1 + \sum_{i=1}^r \xi_{u,v}^{(i)}$ (the expression in (10) is strictly increasing in h_c for $\xi_{u,v}^{(r+1)} > 0$). In particular, for $r = 0$, $h_c \geq 1$, which is obviously true. For $r = 1$ (the second visit), the token has at least made $\xi_{u,v}^{(1)}$ updates, etc., from which the lower bound on h_c follows. ■

C. Proof of Theorem 2

For the Skip-Ring scheme, in every round r (unless all nodes are skipped), there exists a pair of neighboring nodes $(\tilde{u}^{(r)}, \tilde{v}^{(r)})$ for which the token travels directly from $\tilde{u}^{(r)}$ to $\tilde{v}^{(r)}$. Hence, $\xi_{\tilde{u}^{(r)},\tilde{v}^{(r)}}^{(r)} = 1$ for all r, and it follows from Lemma 12 that

$$\max_{u,v\in\mathcal{V},\ u\neq v} \mathscr{D}_\alpha\Big(\mathcal{O}_v^{(r)}(\mathcal{A}(\mathcal{D}))\,\Big\|\,\mathcal{O}_v^{(r)}(\mathcal{A}(\mathcal{D}'))\Big) \leq \frac{2\alpha k^2}{\sigma^2}. \tag{11}$$

The number of visits of the token to a node v during the execution of the algorithm, denoted by Ξ_v, follows a binomial distribution with parameters ${}^{h_{\max}}/_n$ (number of independent trials) and $1-p$ (success probability). Let $\bar{h}$ be defined as in the formulation of the theorem. Then, it follows from a standard Chernoff bound that $\Pr(\Xi_v \geq \bar{h}) \leq \delta'$, for some $\delta' \in (0, 1)$. Now,

$$\begin{aligned} &\max_{u,v\in\mathcal{V},\ u\neq v} \mathscr{D}_\alpha\big(\mathcal{O}_v(\mathcal{A}(\mathcal{D}))\,\big\|\,\mathcal{O}_v(\mathcal{A}(\mathcal{D}'))\big) \\ &\quad \overset{(a)}{\leq} \max_{u,v\in\mathcal{V},\ u\neq v} \sum_{r=1}^{\bar{h}} \mathscr{D}_\alpha\Big(\mathcal{O}_v^{(r)}(\mathcal{A}(\mathcal{D}))\,\Big\|\,\mathcal{O}_v^{(r)}(\mathcal{A}(\mathcal{D}'))\Big) \end{aligned}$$

$$\overset{(b)}{\leq} \sum_{r=1}^{\tilde{h}} \max_{u,v\in\mathcal{V},\ u\neq v} \mathscr{D}_\alpha\left(\mathcal{O}_v^{(r)}(\mathcal{A}(\mathcal{D}))\,\middle\|\,\mathcal{O}_v^{(r)}(\mathcal{A}(\mathcal{D}'))\right)$$

$$\overset{(c)}{\leq} \sum_{r=1}^{\tilde{h}} \frac{2\alpha k^2}{\sigma^2} = \frac{2\alpha k^2}{\sigma^2}\cdot\tilde{h} \tag{12}$$

for every $\alpha > 1$, where $\mathcal{D} \sim_u \mathcal{D}'$. (a) follows from the composition theorem for RDP (Lemma 9) and Lemma 10, (b) from swapping the order of maximization and summation, and (c) from (11).

Then, converting from RDP to DP using Lemma 11 gives that Algorithm 1 satisfies

$$\left(\frac{2\tilde{h}\alpha k^2}{\sigma^2} + \frac{\log(1/\delta)}{\alpha-1}, \delta+\delta'\right) - \text{NDP}. \tag{13}$$

Now, the Rényi divergence parameter α can be optimized in order to minimize $2\tilde{h}\alpha k^2/\sigma^2 + \log(1/\delta)/(\alpha-1)$ by taking the derivative with respect to α. Doing so, gives $\alpha = 1 + \frac{\sigma\sqrt{\log(1/\delta)}}{k\sqrt{2\tilde{h}}} > 1$ from which the result follows by substituting this value of α into (13) and setting $\sigma = \frac{k\sqrt{8\log(1.25/\delta)}}{\varepsilon}$, where $\varepsilon > 0$ and $0 < \delta < 1$.

D. Proof of Theorem 3

In contrast to the proof of Theorem 2, the distance between any pair of two nodes u, v is random over the rounds of the algorithm. Hence, we have to resort to a weak form of convexity for Rényi divergence as formulated in Lemma 7. We start with a technical lemma.

Lemma 13: The fraction $\xi_{u,v}^{(r+1)}/(\varrho_{u,v}^{(r+1)})^2$ from (9) is upper-bounded by 1.

Now, let $\Xi_{u,v}^{(r)}$ denote the actual number of noise terms added in between the $(r-1)$-th and r-th visit of the token at node v after visiting node u. $\Xi_{u,v}^{(r)}$ is a binomial RV with parameters $d^{(r)}(u,v)$ and $1-p$, where $d^{(r)}(u,v)$ is the distance between u and v along the direction of the token over the ring. From Lemma 7, it follows that

$$\mathscr{D}_\alpha\left(\mathcal{O}_v^{(r)}(\mathcal{A}(\mathcal{D}))\,\middle\|\,\mathcal{O}_v^{(r)}(\mathcal{A}(\mathcal{D}'))\right) \leq (1+b)$$
$$\times\mathbb{E}\left[\mathscr{D}_\alpha\left(\mathcal{O}_v^{(r)}(\mathcal{A}(\mathcal{D}))\,\middle\|\,\mathcal{O}_v^{(r)}(\mathcal{A}(\mathcal{D}'))\middle|\Xi_{u,v}^{(i)} = \xi_{u,v}^{(i)}, i\in[r]\right)\right],$$

where $\mathscr{D}_\alpha\left(\mathcal{O}_v^{(r)}(\mathcal{A}(\mathcal{D}))\,\middle\|\,\mathcal{O}_v^{(r)}(\mathcal{A}(\mathcal{D}'))|\Xi_{u,v}^{(i)} = \xi_{u,v}^{(i)}, i\in[r]\right)$ is the Rényi divergence between the views $\mathcal{O}_v^{(r)}(\mathcal{A}(\mathcal{D}))$ and $\mathcal{O}_v^{(r)}(\mathcal{A}(\mathcal{D}'))$ given that in between the $(i-1)$-th and i-th visit of the token at node v, $\xi_{u,v}^{(i)} \in [d^{(i)}(u,v)]$ nodes after node u (including) have been visited, and where $0 < b \leq 1$ is a constant such that

$$\mathscr{D}_\alpha\left(\mathcal{O}_v^{(r)}(\mathcal{A}(\mathcal{D}))\,\middle\|\,\mathcal{O}_v^{(r)}(\mathcal{A}(\mathcal{D}'))\middle|\Xi_{u,v}^{(i)} = \xi_{u,v}^{(i)}, i\in[r]\right)$$
$$\leq \frac{b}{\alpha-1} \tag{14}$$

for all $\xi_{u,v}^{(i)} \in [d^{(i)}(u,v)]$. By picking $b = 1$ and applying Lemma 12, gives the expression in (15) at the bottom of the page. As $\xi_{u,v}^{(r+1)}/(\varrho_{u,v}^{(r+1)})^2 \leq 1$ (see Lemma 13), in order to satisfy (14) (with $b = 1$), we require that $2\alpha(\alpha-1)k^2 \leq \sigma^2$ (see (9)), which is equivalent to $\frac{1-\sqrt{2\frac{\sigma^2}{k^2}+1}}{2} \leq \alpha \leq \frac{1+\sqrt{2\frac{\sigma^2}{k^2}+1}}{2}$. Since the lower bound on α above is less than one,

$$1 < \alpha \leq \frac{1+\sqrt{2\frac{\sigma^2}{k^2}+1}}{2} = \frac{1+\sqrt{\frac{16\log(1.25/\delta)}{\varepsilon^2}+1}}{2}, \tag{16}$$

where we have used that $\sigma = \frac{k\sqrt{8\log(1.25/\delta)}}{\varepsilon}$.

In the following, to simplify notation, let $g(\Xi_{u,v}^{(1)}\ldots,\Xi_{u,v}^{(r)})$ denote the expression inside the expectation operator of (15). It follows that

$$\mathbb{E}_{\Xi_{u,v}^{(1)},\ldots,\Xi_{u,v}^{(r)}}\left[g(\Xi_{u,v}^{(1)}\ldots,\Xi_{u,v}^{(r)})\right]$$
$$= \sum_{d_1=1}^{n-1}\cdots\sum_{d_r=1}^{n-1}\prod_{i=1}^{r}\left[\Pr(d^{(i)}(u,v)=d_i)\right]$$
$$\times\,\mathbb{E}_{\Xi_{u,v}^{(1)},\ldots,\Xi_{u,v}^{(r)}}\left[g(\Xi_{u,v}^{(1)},\ldots,\Xi_{u,v}^{(r)})\;\middle|\;d^{(i)}(u,v)=d_i, i\in[r]\right],$$

where

$$\mathbb{E}_{\Xi_{u,v}^{(1)},\ldots,\Xi_{u,v}^{(r)}}\left[g(\Xi_{u,v}^{(1)},\ldots,\Xi_{u,v}^{(r)})\;\middle|\;d^{(i)}(u,v)=d_i, i\in[r]\right]$$
$$= \sum_{h_1=1}^{d_1}\cdots\sum_{h_r=1}^{d_r} g(h_1,\ldots,h_r)$$
$$\times\binom{d_1}{h_1}\cdots\binom{d_r}{h_r}p^{d_1+\cdots+d_r-(h_1+\cdots h_r)}(1-p)^{h_1+\cdots+h_r}.$$

Now, for a fixed pair of nodes u, v, $d^{(i)}(u,v) = 1$ with probability $1/(n-1)$, $d^{(i)}(u,v) = 2$ with probability $(1-1/(n-1))\cdot 1/(n-2) = 1/(n-1)$, $d^{(i)}(u,v) = 3$ with probability $(1-1/(n-1))\cdot(1-1/(n-2))\cdot 1/(n-3) = 1/(n-1)$, etc. Hence, $d^{(i)}(u,v)$ follows a uniform distribution. As a result,

$$\mathbb{E}_{\Xi_{u,v}^{(1)},\ldots,\Xi_{u,v}^{(r)}}\left[g(\Xi_{u,v}^{(1)},\ldots,\Xi_{u,v}^{(r)})\right]$$
$$= \frac{1}{(n-1)^r}\sum_{d_1=1}^{n-1}\cdots\sum_{d_r=1}^{n-1}\sum_{h_1=1}^{d_1}\cdots\sum_{h_r=1}^{d_r} g(h_1,\ldots,h_r)$$

$$\max_{u,v\in\mathcal{V},\ u\neq v}\mathscr{D}_\alpha\left(\mathcal{O}_v^{(r)}(\mathcal{A}(\mathcal{D}))\,\middle\|\,\mathcal{O}_v^{(r)}(\mathcal{A}(\mathcal{D}'))\right)$$
$$\leq \max_{u,v\in\mathcal{V},\ u\neq v}(1+1)\cdot\frac{2\alpha k^2}{\sigma^2}\mathbb{E}_{\Xi_{u,v}^{(1)},\ldots,\Xi_{u,v}^{(r)}}\left[\frac{\Xi_{u,v}^{(r)}}{4(1+\sum_{i=1}^{r-1}\Xi_{u,v}^{(i)})\cdot\left(\sqrt{1+\sum_{i=1}^{r-1}\Xi_{u,v}^{(i)}+\Xi_{u,v}^{(r)}}-\sqrt{1+\sum_{i=1}^{r-1}\Xi_{u,v}^{(i)}}\right)^2}\right] \tag{15}$$

$$\times \binom{d_1}{h_1} \cdots \binom{d_r}{h_r} p^{d_1+\cdots+d_r-(h_1+\cdots h_r)}(1-p)^{h_1+\cdots+h_r}$$

$$\overset{(a)}{\leq} \frac{1}{(n-1)} \sum_{d=1}^{n-1} \sum_{h=1}^{d} g(h,\ldots,h) \binom{d}{h} p^{d-h}(1-p)^h$$

which is independent of u, v, and where (a) follows from the fact that $g(\cdot,\ldots,\cdot)$ is a decreasing and convex function. Hence,

$$\max_{\substack{u,v\in\mathcal{V},\\ u\neq v}} \mathscr{D}_\alpha\left(\mathcal{O}_v^{(r)}(\mathcal{A}(\mathcal{D})) \,\middle\|\, \mathcal{O}_v^{(r)}(\mathcal{A}(\mathcal{D}'))\right)$$

$$\leq \frac{4\alpha k^2}{(n-1)\sigma^2} \sum_{d=1}^{n-1} \sum_{h=1}^{d} g(h,\ldots,h) \binom{d}{h} p^{d-h}(1-p)^h. \quad (17)$$

As for the `Skip-Ring` scheme, the number of visits of the token to a node v during the execution of the algorithm, denoted by Ξ_v, follows a binomial distribution with parameters ${}^{h_{\max}}/{}_{n}$ and $1-p$. Let $\tilde{h}$ be defined as in the formulation of the theorem. Then, it follows from a standard Chernoff bound that $\Pr(\Xi_v \geq \tilde{h}) \leq \delta'$, for some $\delta' \in (0,1)$. Applying the composition theorem for RDP (Lemma 9), Lemma 10, and swapping the order of maximization and summation as in the derivations in (12), but using (17) together with the definition of $g(\cdot)$ from (15), results in $\max_{u,v\in\mathcal{V},\ u\neq v} \mathscr{D}_\alpha\left(\mathcal{O}_v(\mathcal{A}(\mathcal{D})) \,\middle\|\, \mathcal{O}_v(\mathcal{A}(\mathcal{D}'))\right) \leq \frac{4a\alpha k^2}{\sigma^2}$, where a is defined in the theorem formulation.

Then, converting from RDP to DP using Lemma 11 gives that Algorithm 1 satisfies

$$\left(\frac{4a\alpha k^2}{\sigma^2} + \frac{\log({}^{1}/{}_{\delta})}{\alpha-1}, \delta+\delta'\right) - \text{NDP}, \quad (18)$$

where again the parameter α can be optimized in order to minimize the ε (left) term in (18). However, there is a subtlety as the condition in (16) must be satisfied. Taking the derivative of the ε (left) term of (18) with respect to α, equating it to zero, and setting $\sigma = \frac{k\sqrt{8\log({}^{1.25}/{}_{\delta})}}{\varepsilon}$, where $\varepsilon > 0$ and $0 < \delta < 1$, gives $\alpha = 1 + \frac{\sqrt{2\log({}^{1}/{}_{\delta})\log({}^{1.25}/{}_{\delta})}}{\varepsilon\sqrt{a}} > 1$ and the final result follows by substituting the minimum of the optimal value of α from above and the right-hand-side upper bound of (16) into the ε (left) term of (18) and simplifying.

References

[1] Y. Yakimenka, C.-W. Weng, H.-Y. Lin, E. Rosnes, and J. Kliewer, "Straggler-resilient differentially-private decentralized learning," in *Proc. IEEE Inf. Theory Workshop (ITW)*, 2022, pp. 708–713.

[2] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Agüera y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2017, pp. 1273–1282.

[3] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," in *NeurIPS Workshop Private Multi-Party Mach. Learn. (PMPML)*, 2016, pp. 1–11.

[4] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.

[5] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent," in *Proc. 31th Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2017, pp. 5336–5346.

[6] G. Xiong, G. Yan, R. Singh, and J. Li, "Straggler-resilient distributed machine learning with dynamic backup workers," 2021, *arXiv:2102.06280v1*.

[7] G. Neglia, C. Xu, D. Towsley, and G. Calbi, "Decentralized gradient methods: Does topology matter?" in *Proc. 23rd Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2020, pp. 2348–2358.

[8] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Security (CCS)*, 2015, pp. 1322–1333.

[9] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," 2017, *arXiv:1712.07557v2*.

[10] K. Wei et al., "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3454–3469, 2020.

[11] E. Cyffers and A. Bellet, "Privacy amplification by decentralization," in *Proc. 25th Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2022, pp. 5334–5353.

[12] B. Balle, G. Barthe, and M. Gaboardi, "Privacy amplification by subsampling: Tight analyses via couplings and divergences," in *Proc. 32th Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2018, pp. 6280–6290.

[13] V. Feldman, I. Mironov, K. Talwar, and A. Thakurta, "Privacy amplification by iteration," in *Proc. 59th Annu. IEEE Symp. Found. Comput. Sci. (FOCS)*, 2018, pp. 521–532.

[14] Ú. Erlingsson, V. Feldman, I. Mironov, A. Raghunathan, K. Talwar, and A. Thakurta, "Amplification by shuffling: From local to central differential privacy via anonymity," in *Proc. Annu. ACM-SIAM Symp. Discrete Algorithms (SODA)*, 2019, pp. 2468–2479.

[15] V. Feldman, A. McMillan, and K. Talwar, "Hiding among the clones: A simple and nearly optimal analysis of privacy amplification by shuffling," in *Proc. 62th Annu. IEEE Symp. Found. Comput. Sci. (FOCS)*, 2022, pp. 954–964.

[16] E. Cyffers, M. Even, A. Bellet, and L. Massoulié, "Muffliato: Peer-to-peer privacy amplification for decentralized optimization and averaging," in *Proc. 36th Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2022, pp. 15889–15902.

[17] M. Showkatbakhsh, C. Karakus, and S. Diggavi, "Differentially private consensus-based distributed optimization," 2019, *arXiv:1903.07792v1*.

[18] R. Jin, X. He, and H. Dai, "Decentralized differentially private without-replacement stochastic gradient descent," 2018, *arXiv:1809.02727v3*.

[19] K. Bonawitz et al., "Practical secure aggregation for privacy-preserving machine learning," in *Proc. 24th ACM SIGSAC Conf. Comput. Commun. Security (CCS)*, 2017, pp. 1175–1191.

[20] S. Kadhe, N. Rajaraman, O. O. Koyluoglu, and K. Ramchandran, "FastSecAgg: Scalable secure aggregation for privacy-preserving federated learning," 2020, *arXiv:2009.11248v1*.

[21] J. So et al., "LightSecAgg: A lightweight and versatile design for secure aggregation in federated learning," in *Proc. Conf. Mach. Learn. Syst. (MLSys)*, 2022, pp. 694–720.

[22] J. So, B. Güler, and A. S. Avestimehr, "Turbo-aggregate: Breaking the quadratic aggregation barrier in secure federated learning," *IEEE J. Sel. Areas Inf. Theory*, vol. 2, no. 1, pp. 479–489, Mar. 2021.

[23] R. Schlegel, S. Kumar, E. Rosnes, and A. Graell i Amat, "CodedPaddedFL and CodedSecAgg: Straggler mitigation and secure aggregation in federated learning," *IEEE Trans. Commun.*, vol. 71, no. 4, pp. 2013–2027, Apr. 2023.

[24] A. Reisizadeh, H. Taheri, A. Mokhtari, H. Hassani, and R. Pedarsani, "Robust and communication-efficient collaborative learning," in *Proc. 33th Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2019, pp. 8388–8399.

[25] Z. Charles and J. Konečný, "On the outsized importance of learning rates in local update methods," 2020, *arXiv:2007.00878v1*.

[26] A. Mitra, R. Jaafar, G. J. Pappas, and H. Hassani, "Linear convergence in federated learning: Tackling client heterogeneity and sparse gradients," in *Proc. 35th Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2021, pp. 14606–14619.

[27] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran, "Speeding up distributed machine learning using codes," *IEEE Trans. Inf. Theory*, vol. 64, no. 3, pp. 1514–1529, Mar. 2018.

[28] S. Li and S. Avestimehr, "Coded computing: Mitigating fundamental bottlenecks in large-scale distributed computing and machine learning," *Found. Trends® Commun. Inf. Theory*, vol. 17, no. 1, pp. 1–148, Aug. 2020.

[29] Q. Yu, "Coded computing: A transformative framework for resilient, secure, private, and communication efficient large scale distributed computing," Ph.D. dissertation, Electr. Eng., Univ. Southern California, Los Angeles CA, USA, Aug. 2020.

[30] A. Severinson, A. Graell i Amat, and E. Rosnes, "Block-diagonal and LT codes for distributed computing with straggling servers," *IEEE Trans. Commun.*, vol. 67, no. 3, pp. 1739–1753, Mar. 2019.

[31] S. Dutta, V. Cadambe, and P. Grover, "“Short-dot": Computing large linear transforms distributedly using coded short dot products," *IEEE Trans. Inf. Theory*, vol. 65, no. 10, pp. 6171–6193, Oct. 2019.

[32] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "Polynomial codes: An optimal design for high-dimensional coded matrix multiplication," in *Proc. 31th Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2017, pp. 4403–4413.

[33] S. Dutta, M. Fahim, F. Haddadpour, H. Jeong, V. Cadambe, and P. Grover, "On the optimal recovery threshold of coded matrix multiplication," *IEEE Trans. Inf. Theory*, vol. 66, no. 1, pp. 278–301, Jan. 2020.

[34] Q. Yu and A. S. Avestimehr, "Coded computing for resilient, secure, and privacy-preserving distributed matrix multiplication," *IEEE Trans. Commun.*, vol. 69, no. 1, pp. 59–72, Jan. 2021.

[35] W.-T. Chang and R. Tandon, "On the capacity of secure distributed matrix multiplication," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, 2018, pp. 1–6.

[36] J. Kakar, S. Ebadifar, and A. Sezgin, "On the capacity and straggler-robustness of distributed secure matrix multiplication," *IEEE Access*, vol. 7, pp. 45783–45799, 2019.

[37] H. Yang and J. Lee, "Secure distributed computing with straggling servers using polynomial codes," *IEEE Trans. Inf. Forensics Security*, vol. 14, pp. 141–150, 2019.

[38] M. Aliasgari, O. Simeone, and J. Kliewer, "Distributed and private coded matrix computation with flexible communication load," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2019, pp. 1092–1096.

[39] R. G. L. D'Oliveira, S. El Rouayheb, and D. Karpuk, "GASP codes for secure distributed matrix multiplication," *IEEE Trans. Inf. Theory*, vol. 66, no. 7, pp. 4038–4050, Jul. 2020.

[40] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "Straggler mitigation in distributed matrix multiplication: Fundamental limits and optimal coding," *IEEE Trans. Inf. Theory*, vol. 66, no. 3, pp. 1920–1933, Mar. 2020.

[41] N. Mital, C. Ling, and D. Gündüz, "Secure distributed matrix computation with discrete Fourier transform," *IEEE Trans. Inf. Theory*, vol. 68, no. 7, pp. 4666–4680, Jul. 2022.

[42] R. Tandon, Q. Lei, A. G. Dimakis, and N. Karampatziakis, "Gradient coding: Avoiding stragglers in distributed learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2017, pp. 3368–3376.

[43] C. Karakus, Y. Sun, S. Diggavi, and W. Yin, "Straggler mitigation in distributed optimization through data encoding," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2017, pp. 5440–5448.

[44] C.-S. Yang, R. Pedarsani, and A. S. Avestimehr, "Timely-throughput optimal coded computing over cloud networks," in *Proc. 20th ACM Int. Symp. Mobile Ad Hoc Netw. Comput. (MobiHoc)*, 2019, pp. 301–310.

[45] Q. Yu, S. Li, N. Raviv, S. M. M. Kalan, M. Soltanolkotabi, and A. S. Avestimehr, "Lagrange coded computing: Optimal design for resiliency, security, and privacy," in *Proc. 22nd Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2019, pp. 1215–1225.

[46] R. Bitar, M. Wootters, and S. El Rouayheb, "Stochastic gradient coding for straggler mitigation in distributed learning," *IEEE J. Sel. Areas Inf. Theory*, vol. 1, no. 1, pp. 277–291, May 2020.

[47] H.-P. Wang and I. Duursma, "Parity-checked Strassen algorithm," 2020, *arXiv:2011.15082v3*.

[48] C.-S. Yang and A. S. Avestimehr, "Coded computing for secure Boolean computations," *IEEE J. Sel. Areas Inf. Theory*, vol. 2, no. 1, pp. 326–337, Mar. 2021.

[49] J. Kosaian, K. V. Rashmi, and S. Venkataraman, "Learning-based coded computation," *IEEE J. Sel. Areas Inf. Theory*, vol. 1, no. 1, pp. 227–236, May 2020.

[50] A. R. Elkordy, S. Prakash, and S. Avestimehr, "Basil: A fast and Byzantine-resilient approach for decentralized training," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 9, pp. 2694–2716, Sep. 2022.

[51] Z. Wang, Y. Hu, J. Xiao, and C. Wu, "Efficient ring-topology decentralized federated learning with deep generative models for industrial artificial intelligent," 2021, *arXiv:2104.08100v1*.

[52] O. Shamir and T. Zhang, "Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2013, pp. 71–79.

[53] "UCI housing dataset." OpenML. Oct. 1, 2023. [Online]. Available: https://www.openml.org/d/823

[54] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[55] A. Krizhevsky, "Learning multiple layers of features from tiny images," Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, Rep. TR-2009, Apr. 2009.

[56] A. Severinson, E. Rosnes, S. El Rouayheb, and A. Graell i Amat, "DSAG: A mixed synchronous-asynchronous iterative method for straggler-resilient learning," *IEEE Trans. Commun.*, vol. 71, no. 2, pp. 808–822, Feb. 2023.

[57] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Local privacy and statistical minimax rates," in *Proc. 54th Annu. IEEE Symp. Found. Comput. Sci. (FOCS)*, 2013, pp. 429–438.

[58] S. Dutta, V. Cadambe, and P. Grover, "Coded convolution for parallel and distributed computing within a deadline," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2017, pp. 2403–2407.

[59] G. Neglia, G. Calbi, D. Towsley, and G. Vardoyan, "The role of network topology for distributed machine learning," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, 2019, pp. 2350–2358.

[60] T. Sun, Y. Sun, and W. Yin, "On Markov chain gradient descent," in *Proc. 32th Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2018, pp. 9918–9927.

[61] G. Ayache and S. El Rouayheb, "Private weighted random walk stochastic gradient descent," *IEEE J. Sel. Areas Inf. Theory*, vol. 2, no. 1, pp. 452–463, Mar. 2021.

[62] I. Mironov, "Rényi differential privacy," in *Proc. 30th IEEE Comput. Security Found. Symp. (CSF)*, 2017, pp. 263–275.

[63] A. Koloskova, N. Loizou, S. Boreiri, M. Jaggi, and S. U. Stich, "A unified theory of decentralized SGD with changing topology and local updates," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2020, pp. 5381–5393.

[64] Z. Chen, M. Dahl, and E. G. Larsson, "Decentralized learning over wireless networks: The effect of broadcast with random access," in *Proc. IEEE Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, 2023, pp. 316–320.

[65] D. P. Herrera, Z. Chen, and E. G. Larsson, "Faster convergence with less communication: Broadcast-based subgraph sampling for decentralized learning over wireless networks," 2024, *arXiv:2401.13779v1*.

[66] S. De, L. Berrada, J. Hayes, S. L. Smith, and B. Balle, "Unlocking high-accuracy differentially private image classification through scale," 2022, *arXiv:2204.13650v2*.

[67] S. Hu, X. Yuan, W. Ni, X. Wang, E. Hossain, and H. V. Poor, "OFDMA-F^{2}L: Federated learning with flexible aggregation over an OFDMA air interface," *IEEE Trans. Wireless Commun.*, early access, Jan. 15, 2024, doi: 10.1109/TWC.2023.3334691.

[68] L. G. Gubin, B. T. Polyak, and E. V. Raik, "The method of projections for finding the common point of convex sets," *USSR Comput. Math. Math. Phys.*, vol. 7, no. 6, pp. 1–24, Jan. 1967.

[69] T. van Erven and P. Harremoës, "Rényi divergence and Kullback-Leibler divergence," *IEEE Trans. Inf. Theory*, vol. 60, no. 7, pp. 3797–3820, Jul. 2014.