Efficient Communication Topology via Partially Differential Privacy for Decentralized Learning

Cheng-Wei Ching[†§], Hung-Sheng Huang[†§], Chun-An Yang[†], Yu-Chun Liu[‡], and Jian-Jhih Kuo[†*]

***Abstract*—Decentralized learning (DL) allows IoT devices to exchange local model updates with only their neighboring devices instead of sending their model updates to a central server for aggregation. However, current DL frameworks cannot support the emerging Social IoT (SIoT) paradigm since SIoT devices exchange model updates with only social neighbors based on specific social relations (e.g., ownership and parental relationships). Conversely, sharing model updates with non-social neighbors can improve training performance but may violate social relations. Differential privacy (DP) is thus engaged with DL to ensure data security, while excessive devices engaging DP may downgrade the training performance. However, most research neglects the effect of neighbor selection for each device based on social networks, physical networks, and DP. Therefore, in this paper, we explore the non-trivial relation among the above factors to present a DL framework, DeepPrivacy, and prove its convergence rate and DP. Then, we formulate a novel optimization problem, CoTOPO, to find an efficient communication topology[1] for model updates exchange among devices in DL, and propose an algorithm, AutoTag, for CoTOPO. Last, experiment results manifest that DeepPrivacy and AutoTag combined outperform the state of the art in terms of convergence rate and physical training time significantly on CIFAR10 and FMNIST.**

***Index Terms*—Social Internet-of-Things, Decentralized Learning, Communication Topology, Partially Differential Privacy**

I. Introduction

Recently, Social Internet of Things (SIoT) with *Artificial Intelligence (AI) on chips* is a promising network paradigm, where a set of SIoT devices monitor the environment and collect the data (e.g., photos, voices, positions, signal), and interact and establish relationship with each other to tackle a specific task [1]. The SIoT devices can build *parental object relation* and *ownership object relation* if they have the same manufacturers and owners, respectively [1], [2]. However, due to privacy issues, collecting and uploading *private* data from SIoT devices to the central server with intensive computing power for training are impracticable. Federated Learning (FL) is thus designed to deal with the issue [3], where each device processes its private data *locally* and uploads *only* the parameter updates to a central server for aggregation. Nevertheless, the communication bottleneck arises when FL is adopted. The central server suffers from receiving the surge of parameter updates with limited bandwidth [4] and prolongs the training.

To overcome the communication bottleneck, decentralized learning (DL) is proposed, where each participant (i.e., SIoT device) execute *on-device* training but exchange parameter updates with its *neighbors* according to a given *communication topology*.[1] In this sense, each device acts as both a training unit and a parameter aggregator at the same time, and the role of the central parameter server in FL is no longer needed [5]. The *SIoT platforms* can take an overview of *social network* of devices [6], [7] so they are suitable to arrange the communication topologies for devices. For example, iSapiens is an SIoT-enabled platform featuring *relationship and trustworthiness managements* among devices. The features facilitate the assessment and management of social relations in the SIoT paradigm, where the relations are typically established *autonomously* according to prespecified social conditions [6]. Associated to iSapiens, devices and their social profiles are registered in the platform. However, most research focuses on data compression [5] but ignores the potential communication time caused by physical networks. Then, a crucial bottleneck arises in *synchronous* procedures of DL and prolongs training time if the exploited links have long paths between devices in the physical networks [8]. The fast-growing process power and the slow-developing communication technology exacerbate the issue. For instance, the process power of Nvidia GPUs has increased by 30x in the last 10 years while the network speed has only increased by 20x (4G to 5G) in the last 10 years [8].

Executing DL with communication topologies composed of *only* social links between devices (e.g., ownership or parental relations) in a sparse social network may limit the convergence rate and prolong the training procedure [9]. In contrast, exploiting extra non-social links (i.e., exchange parameters with strangers) can speed up the training but may put user data at risk since user data may be reproduced unknowingly [10]. Fortunately, differential privacy (DP) engaged with devices to add noises in the model updates for exchange can mitigate the security issues [10], [11]. Thus, a non-social link can be exploited if its two endpoint devices are both DP devices (i.e., devices engaging DP). Conversely, exploiting DP devices excessively may degrade the convergence rate [10]–[12]. However, selecting a reasonable number of DP devices to exploit additional non-social links to accelerate training while retaining the side effect of employing DP has not been studied.

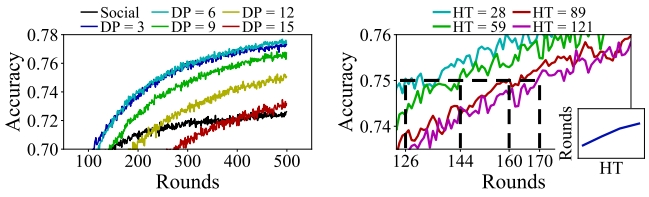
To verify the above two issues, we conduct *two motivating experiments* regarding SIoT-based DL as follows. The structure of adopted neural network for testing comprises 2 convolutional layers (CL) followed by 3 fully connected layers (FCL) to learn a classification task for classifying 10 objects in CIFAR10 [13] (see Section VI for more the implementation details) and the *synchronous updating rules* follow the configuration in [14]. To delve into the impacts posed by *the added noises* and *correspondingly constructed communication topologies*, we extracted random *16 devices* with physical and

[†]Dept. of Computer Science, National Chung Cheng University, Taiwan
[‡]Dept. of Electrical Engineering, National Chung Cheng University, Taiwan
* indicates the corresponding author; [§] denotes the equal contributions.
Corresponding author's email: lajacky@cs.ccu.edu.tw

[1]The communication topology is a virtual network that indicates *logical* connectivity among participants. The logical connectivity can be established according to specific rules. Two devices are neighbors *iff* there exists a link between them and they will exchange model updates during DL process. FL does not need communication topologies since each participant uploads model updates onto the parameter server and suffers from communication bottlenecks.

TABLE I
TRAINING TIME IN DIFFERENT PHYSICAL NETWORK (16 DEVICES)

Target Accuracy	65%	68%	71%	73%
Number of Rounds	69	101	202	580
Computation Time (min)	131.1	191.4	383.2	1101.3
Communication Time in Net1 (min)	43.3	63.4	126.7	363.9
Communication Time in Net2 (min)	10.8	15.8	31.7	90.8



(a) Accuracy vs # DP devices (b) Accuracy vs hitting time (HT)

Fig. 1. (a) Effect of number of DP devices ranging $[0, 16]$ on accuracy and (b) effect of hitting time ranging $[28, 121]$ on accuracy in a 16-device network.

social links (ownership object relations) among them from Santander (a city in Spain) [15] (see Section VI for more details regarding the dataset Santander). We assume that the devices are equipped with *the same computing capabilities* (i.e., the same amount of time to finish locally one-round training in DL), and the size of the neural network for exchange among devices is approximately 4MB.[2] First, the effects of different physical networks on communication time are shown in Table I. The two physical networks extracted from Santander, Net1 and Net2, have the same number of devices (i.e., 16) and take the identical social network structures as the communication topology. Still, they have *different communication bottlenecks* of 37.6 and 9.4 sec, respectively, derived from different LTE-M releases [19].[3] The results show that the training time in different physical networks can be deeply influenced (i.e., $363.9 - 90.8 \approx 273$ min as the target accuracy is 73%) by the communication bottleneck. Likewise, Fig. 1(a) shows the performance achieved by *adopting different number of DP devices* based on the same social network as the communication topology. Intuitively, the more DP devices are adopted, the more links can be included in the communication topology. The convergence rate with only social links is much worse than that with extra non-social links, but that with more than six DP devices will decrease, implying that more noises may prolong the process. Thus, the physical training time (i.e., the time required to get a target accuracy) for DL depends on not only communication topologies and physical networks but the number of DP devices. However, the subtle relation among the physical networks, social networks, and DP and their effects on convergence rate have not been explored jointly to build a good *communication topology* with non-social and social links to reduce the physical training time.

Optimizing the *physical training time* by selecting a suitable set of DP devices and a set of social and non-social links to build a communication topology for DL has new challenges: 1) *Trade-off between global and local iterates*. Global iterate is the number of rounds to achieve converged accuracy for a given task learned in decentralized fashion, and local iterate is the time elapsed for devices to finish parameter exchange and local gradient descent in a round. Intuitively, exploiting more links in the communication topology can increase the connectivity and help reduce the global iterate. However, it may make devices more distant to their neighbors and cause higher local iterate, which may prolong physical training time. The constraint of the social relations among devices further exacerbates the issue. 2) *Uncertain global iterate*. A smaller global iterate helps reduce the physical training time. Nevertheless, adding more links does not always imply a smaller global iterate, and it is difficult to explicitly define global iterate [20]. Fortunately, by [9], we can infer that the global iterate of a given communication topology is asymptotically proportional to the *hitting time*.[4] (see Definition 6), which highly depends on topology structure. The effect of hitting time on convergence (with no DP noise) is shown in Fig. 1(b). Clearly, a smaller global iterate comes from a lower hitting time and their relation is almost *linear*. 3) *Varying relation between global iterate and DP devices*. Exploiting extra non-social links achieves lower global iterate at first since the sparsity of communication topology is reduced. However, it may come at the expense of more rounds to achieve the required accuracy since the noises of DP devices increases accordingly. Thereby, besides the hitting time, the ratio of DP devices (i.e., ratio of devices engaging DP in the network) also holds sway on global iterate. In summary, given a social network and a physical network, it is challenging to judge if physical training time can be further reduced by adding more DP devices or explicitly using *available* links (i.e., social links or non-social links between existing DP devices). It is because both manners subtly influence the growth and decline of physical training time.

To address the above challenges jointly, we first propose the **De**centralized Optimization Fram**e**work with **P**artially Differential **Privacy** (DeepPrivacy), which is run in the SIoT platforms and arranges a subset of devices to engage DP so as to exploit extra non-social links *for training acceleration*. To fully utilize DeepPrivacy, we present a new optimization problem named **Co**nstruction of Time-Efficient Communication **Topo**logy in Social Networks for DeepPrivacy (CoTOPO) as follows. With the given parameters: 1) a social network and 2) a physical network with node and link weights (i.e., communication and computation time[5]), CoTOPO asks for a set of links to construct a communication topology that yields the minimum *physical training time*. An **A**daptive D**u**al-Factor **To**pology Cons**t**ruction **Alg**orithm (AutoTag) is then designed to address the above three challenges to accelerate training.

The novelty and contributions are summarized as follows.

[2]Generally, the SIoT devices are equipped with limited computing power so the structure of neural network for training in the SIoT devices should be relatively lightweight, such as MobileNet [16], and SqeezeNet [17]. Such lightweight models should not be compressed or quantized, or the performance (e.g., the final accuracy or loss) will be drastically downgraded [18].

[3]Note that two devices with a social link may use different bandwidth for communication (e.g., Cat-M1 and Cat-M2) in the physical network, leading to the straggler problem in exchanging model updates. Thus, such low-data-rate links are undesired in the communication topology.

[4]Hitting time can be informally regarded as the expected number of rounds needed to propagate a message from a device v_1 to another device v_2 in the network. Note that the hitting time from v_1 to v_2 can be different to that from v_2 to v_1. The detailed definition is presented in Definition 6 with Example 2.

[5]Note that the computation time for training can be predicted by considering required numbers of flops for traversing training models and the available flops of training devices [21]. The communication time can also be predicted since LTE-M offers expected data rates [22].

- To the best of our knowledge, *this paper makes the first attempt to explore the relation among the social and physical networks and DP in DL for the SIoT devices* and proposes an algorithm, AutoTag, from the perspective of SIoT platforms to build training-time-efficient communication topologies.
- We present a new DL framework, DeepPrivacy, to exploit the communication topologies with non-social and social links constructed by AutoTag and rigorously show the convergence rate of DeepPrivacy for *non-convex* loss functions.
- We evaluate the performance of DeepPrivacy with two well-known datasets, CIFAR10 [13] and FMNIST [23]. The results show that DeepPrivacy outperforms the state-of-the-art by at least 20% of physical training time.

II. Related Work

A. Communication-Efficient Federated Learning (FL)

FL is devised to train the global model locally via multiple user devices with their data to avoid direct data access [3]. To derive the global model for the next-round learning, FL requires a *parameter server* to coordinate and aggregate the local model updates from user devices. Konečnỳ *et al.* present a method to optimize the transmission efficiency between the central server and devices by lightening transmitted data sizes [24]. Wang *et al.* analyze the convergence rate via distributed gradient descent to achieve the best trade-off between the local training epoch and global aggregation [25]. Nishio *et al.* maximize the number of selected devices for a round in FL [26]. However, FL still struggles over the communication bottleneck when the local model updates from the user devices are sent to the parameter server for aggregation.

B. Communication-Efficient Decentralized Learning (DL)

DL is first innovated by Tsitsiklis *et al.* [27], and it is also called gossip algorithm. Different from FL, DL does not require a parameter server to aggregate the local model updates from user devices. Li *et al.* develop a pipelined framework which allows two consecutive computing iterations to overlap on the timeline and mask the faster of the computation and communication to reduce the training time [28]. Koloskova *et al.* propose CHOCO-SGD that quantizes the model updates (e.g., from 32-bit float point to 8-bit integer) and show that CHOCO-SGD achieves linear speedup of convergence rate in the number of training devices compared to SGD on a single node for high compression ratios on general non-convex functions, and non-IID training data [5]. However, none of them considers the construction of communication topologies and the interplay between the social and physical networks.

C. FL and DL with Differential Privacy (DP)

For FL, DP is adopted to add noises into model updates on the device side before aggregation [29]. Arachchige *et al.* propose a promising mechanism where convolutional and fully connected layers are trained on devices and a central server, respectively, and DP is employed to add noises into feature before intermediate data leave devices [30]. For DL, Li *et al.* add noises into parameters before exchanging them with their neighbors [12]. Zhang *et al.* combine the techniques of sparsification with DP to guarantee the data privacy and reduce the model size for exchange in DL [31]. Nevertheless, none of them considers social networks and partially DP mechanism.

III. Preliminaries

A. Decentralized Learning (DL)

We consider a network including a set of SIoT devices V, which are unwilling to share information with unknown devices. A training task is launched in DL fashion of the form

$$f(x) := \frac{1}{|V|}\sum_{i\in V} f_i(x), \qquad \forall x \in \mathcal{X} \subseteq \mathbb{R}^d \tag{1}$$

where $f, f_i : \mathcal{X} \to \mathbb{R}$ are global and local functions, respectively. Each device $i \in V$ has its local function f_i and the local function is usually approximated by *stochastic gradient descent* (SGD) with device i's local data $\mathcal{D}_i$, yielding

$$f_i(x) := \mathbb{E}_{\xi_t \in \mathcal{D}_i} F_i(x, \xi_t), \tag{2}$$

where $F_i : \mathbb{R}^d \times \xi_t \to \mathbb{R}$ denotes the local function approximated by a fraction of data ξ_t *randomly drawn* from $\mathcal{D}_i$ at round t. Here we assume $F_i(\cdot,\cdot)$ is *non-convex*, which is more practical in the machine learning context [5]. Each device *synchronously* updates the local parameter as follows. The devices execute the optimization techniques (e.g., SGD) with local data iteratively to obtain local parameters in each round. To acquire more accurate and precise model parameters, the devices necessitate exchanging local parameters with its neighboring devices. The prespecified communication topology $G_c = (V, E_c)$ determines the connections among the devices so the local parameters are updated *synchronously* as follows.

$$x_i^{t+1} = \sum_{j\in V} a_{ij}(x_j^t + \eta_t g_j^t), \tag{3}$$

where η_t and g_j^t are the learning rate at round t and gradient of device j at round t, respectively, and a_{ij} denotes the entry at the i^{th} row and the j^{th} column of communication matrix $\mathcal{A}(G_c)$ (see Definition 1). Note that each device will receive at least one copy and at most $|V|-1$ copies of local parameters from its neighbors in G_c. The number of received copies depends on the prespecified neighbors. Generally speaking, the less deviated number of the copies each device receives in each round (i.e., the communication topology is less irregular), the better the training results can be. To aggregate the multiple copies, the devices follow the instructions in the communication matrix introduced in Definition 1 to aggregate the copies.

Definition 1 (Lazy-Metropolis-based Communication Matrix [9])**.** Given a set of devices V, the entries a_{ij} of the communication matrix $\mathcal{A}(G_c) \in [0,1]^{|V|\times|V|}$ of communication topology $G_c = \{V, E_c\}$ are defined as

$$a_{ij} = \begin{cases} 1 - \sum_{k\in V\setminus\{i\}} a_{ik}, & \text{if } i = j, \\ \frac{1}{2\max\{deg_c(i), deg_c(j)\}}, & \text{else if } (i,j) \in E_c, \\ 0, & \text{otherwise,} \end{cases} \tag{4}$$

where $deg_c(i)$ denotes the degree of device i in G_c. By (4), the sum over any row or column in $\mathcal{A}(G_c)$ is equal to 1 and $\mathcal{A}(G_c)$ is symmetric so $\mathcal{A}(G_c)$ is a doubly stochastic matrix.

The goal of each device is to find the optimal model parameters $x^* \in \mathbb{R}^d$ such that

$$f(x^*) = \min_{x \in \mathbb{R}^d} \frac{1}{|V|} \sum_{i \in V} f_i(x). \tag{5}$$

There are two stopping criteria. One is to terminate the training process when the prespecified target round, say, 500, is met. The other is to check the performance by averaging over the results of each device (e.g., the averaged accuracy for an object classification task) and decides whether to go through another round of training again to get better performance.

B. Differential Privacy (DP)

Definition 2 (Differential Privacy [11])**.** A randomized algorithm $\mathbb{A} : \mathcal{D} \rightarrow \mathcal{R}$ with domain $\mathcal{D}$ and range $\mathcal{R}$ is said to be (ϵ, δ)-differentially private if for any two adjacent datasets $D, D' \in \mathcal{D}$ that differ on a single data point and for any subset of outputs $S \subseteq \mathcal{R}$, the following inequality holds

$$Pr[\mathbb{A}(D) \in S] \leq e^{\epsilon} Pr[\mathbb{A}(D') \in S] + \delta, \tag{6}$$

where $\epsilon > 0$ and $\delta \in (0, 1)$ are the *privacy budgets*.

Simply put, ϵ and δ should be kept low if the privacy level is highly demanded. However, higher privacy level sacrifices the accuracy of optimization problem (i.e., eq. (1)). Therefore, we need a factor, sensitivity (see Definition 3), to determine how much noise should be generated to perturb the process of optimization and guarantee the privacy level at the same time.

Definition 3 (l_2-Sensitivity [32])**.** Following Definition 2, The sensitivity of a randomized algorithm $\mathbb{A}$ at round $t \geq 0$ (denoted by $\mathbb{A}_t$) is defined as follows

$$\Delta_t = \sup_{D, D' \in \mathcal{D}} \|\mathbb{A}_t(D) - \mathbb{A}_t(D')\|_2. \tag{7}$$

l_2-Sensitivity is important to determine how much noise should be added to guarantee a given privacy level at round t. If Δ_t is higher, we will prefer to add more noises since it could be easy to distinguish between D and D'.

IV. The DL Framework & Optimization Problem

A. The Design of DeepPrivacy for DL

The pseudocode of DeepPrivacy is presented in Algorithm 1. DeepPrivacy requires the initial model parameters x_i^0 for each device i, time-varying learning rate η_t, privacy budget ϵ, convergence index ε, target global round T, and checkpoint round H (see Section VI for more details). Also, DeepPrivacy requires a communication topology G_c which includes 1) the set of devices V and 2) a set of E_c to specify the neighbors for each device in V to exchange local model updates. Note that different communication topologies lead to different convergence rates and thus a novel problem is introduced to optimize the communication topology for DeepPrivacy in Section IV-B. For ease of reading, we go through the statements in DeepPrivacy as shown in Algorithm 1. For each device i, a fraction of training data ξ_i^t is randomly drawn from local dataset $\mathcal{D}_i$ (line 4), and the gradient g_i^t is computed (line 5). Then, if a device is *appointed* to adopt DP mechanism (line 6), it generates noises w_i^t (line 9) based on the sensitivity (lines 7 and 8) to perturb its local gradient, and caches the perturbed parameters (line 10). The devices cache the parameters without perturbing (line 12) if not appointed to adopt DP mechanism (line 11). The devices exchange parameters with the neighbors (line 13) and update local parameters by aggregating local and neighbors' parameters (lines 14). The performance is periodically examined (i.e., once per H rounds) (line 16) by averaging over all the results[6] from each device R_i^t at the t^{th} round (line 17). If it is converged (line 18) or the target round has been achieved (line 2), the training process finishes. We first prove the DP of DeepPrivacy in Proposition 1 and defer the proof to convergence of DeepPrivacy until Section V-A.

Algorithm 1 DeepPrivacy

Input: The initial model parameters x_i^0 for each device $i \in V$, communication topology $G_c = \{V, E_c\}$, communication matrix $\mathcal{A}(G_c)$, time-varying learning rate η_t, convergence index ε, the target round T, the checkpoint round H, privacy budget ϵ, δ.

1: $t \leftarrow 0, Conv \leftarrow false$;
2: **while** (!$Conv$ and $t < T$) **do**
3: **for all** device $i \in V$ **do** *in parallel*
4: Sample ξ_i^t from $\mathcal{D}_i$;
5: Compute gradient $\mathrm{g}_i^t \leftarrow \nabla F_i(x_i^t, \xi_i^t)$;
6: **if** DP mechanism is on **then**
7: Compute sensitivity $\Delta_t \leftarrow 2\eta_t \|g_i^t\|$;
8: $\phi_t \leftarrow \frac{\Delta_t \sqrt{(2 \log 1.25)/\delta}}{\epsilon}$;
9: Generates noises $w_i^t \leftarrow \mathcal{N}(0, (\phi_t)^2)$;
10: $y_i^t \leftarrow x_i^t + \eta_t \mathrm{g}_i^t + w_i^t$;
11: **else**
12: $y_i^t \leftarrow x_i^t + \eta_t \mathrm{g}_i^t$;
13: Send y_i^t and receive y_j^t to/from device j if $(i, j) \in E_c$;
14: $x_i^{t+1} \leftarrow \sum_{j \in V} a_{ij} y_j^t$;
15: **end for**
16: **if** $t \ mod \ H == 0$ **then**
17: Average performance $\mathcal{R}_t = \frac{1}{|V|} \sum_{i \in V} R_i^t$;
18: **if** $t \neq 0$ and $|\mathcal{R}_t - \mathcal{R}_{t-H}| \leq \varepsilon$ **then**
19: $Conv \leftarrow true$;
20: **end if**
21: **end if**
22: $t \leftarrow t + 1$;
23: **end while**

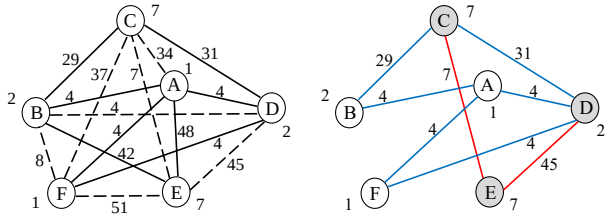
Proposition 1. For each pair of devices that do not have social link in E_s but have to exchange parameters with each other based on E_c, DeepPrivacy achieves (ϵ, δ)-differential privacy.

Due to the page limit, Proposition 1's proof is shown in [33].

B. Problem Formulation of CoTOPO

For user privacy, the communication topology G_c should contain *only* the social links included in the given social network, denoted as $G_s = (V, E_s)$, while the convergence rate is usually compromised crucially due to the *sparsity* and *irregularity* of social network. Thus, DeepPrivacy is innovated to exploit non-social links to accelerate the convergence of DL. However, user data may be unknowingly reproduced [10].

[6]The metrics may be accuracy or loss, depending on the target task.



(a) Social and physical networks (b) A communication topology

Fig. 2. (a) Solid and dashed lines represent social and non-social links in the social network G_s, respectively. The numbers next to each link and node denote the communication time between two devices and computation time of device in physical network G_p, respectively. (b) An example of communication topology G_c. The blue and red links represent the social and non-social links selected in G_c, respectively, and the gray nodes denote the DP devices.

Specifically, the device for DL can receive the shared model updates straight from neighboring unknown devices when the corresponding non-social links between it and the other devices (i.e., non-existent links in E_s) are exploited to exchange model updates and accelerate the training. The receiver may further infer some sensitive information of unknown devices based on the received share model updates [10].

Then, it is necessary for devices to adopt DP mechanism if non-social links are exploited in the communication topology for DL. Exploiting non-social links with DP is usually beneficial to sparse and irregular social networks but more rounds to a specific accuracy may derive in return — the neighbors of DP devices would also suffer lower accuracy since a share of model updates comes with noises. Moreover, exploiting a link between two distant devices (i.e., taking long time to exchange model updates) in the physical network,[7] denoted by $G_p = (V, E_p)$, *may dominate the one-round communication time for exchange* in physical training time. Thus, DeepPrivacy requires a suitable communication topology balancing *global iterate* (i.e., the number of rounds to achieve a specific accuracy) and *local iterate* (i.e., the time required for each round) for computing and exchanging local model updates among devices to reduce physical training time. The link in G_c requiring the most time in G_p will be the physical bottleneck to prolong overall training procedure, so we have Definition 4 as follows.

Definition 4 (Local Iterate)**.** Let r_i and d_{ij} denote the computation time of device $i \in V$ and the communication time between devices $i, j \in V$ in one round. The local iterate is defined as the maximum *communication and computation time* of the links E_c in communication topology G_c, i.e.,

$$\mathcal{L}(G_c) = \max_{(i,j)\in E_c} (d_{ij} + \max\{r_i, r_j\}). \tag{8}$$

Example 1. This example shows how to count local iterate. The social network G_s and physical network G_p with communication time d_{ij} and computation time r_i are shown in Fig. 2(a). Take the topology in Fig. 2(b) for example. By eq. (8), link $\overline{DE}$ dominates the local iterate, which is $45+7=52$. ■

However, explicitly defining global iterate, which we denote by $\mathcal{G}(G_c)$, is non-trivial [20] and will be discussed in the next section. The problem CoTOPO is then defined as follows.

[7]Notice that the physical links between devices may be heavy-weighted since the social relations between devices are not geographically restricted.

TABLE II
ENTRIES m_{ij} IN HITTING TIME MATRIX $\mathcal{M}(G_c)$

m_{ij}	$j=A$	$j=B$	$j=C$	$j=D$	$j=E$	$j=F$
$i=A$	0	15	16.5	10.5	25.3	15.9
$i=B$	11.3	0	11.3	13.5	23.6	23.6
$i=C$	16.5	15	0	10.5	15.9	25.3
$i=D$	13	19.7	13	0	18.8	18.8
$i=E$	18.4	20.5	9	9.5	0	26
$i=F$	9	20.5	18.4	9.5	26	0

Definition 5 (CoTOPO)**.** Given a connected social network $G_s = (V, E_s)$ and a connected weighted physical network $G_p = (V, E_p)$, **Co**nstruction of Time-Efficient Communication **Topo**logy in Social Networks for DeepPrivacy (CoTOPO) asks for a set of links $E_c \subseteq (V \times V)$ to construct a communication topology $G_c = (V, E_c)$ for DeepPrivacy according to Definitions 1 and 4, and minimize *physical training time*, i.e.,

$$\text{minimize} \quad \mathcal{G}(G_c) \cdot \mathcal{L}(G_c). \tag{9}$$

V. ALGORITHM DESIGN — AUTOTAG

We first derive the *critical factors* dominating the global iterate to obtain a regression function to predict the non-trivial global iterate. Then, we design an algorithm to construct multiple candidates of communication topologies and then choose the best candidate via the regression function.

A. Predict the Global Iterate

To overcome the uncertain global iterate, we start by exploring the relations among global iterate, communication topology G_c, and communication matrix $\mathcal{A}(G_c)$. Let $\lambda_i(\mathcal{A}(G_c))$ denote the i^{th} largest eigenvalue of matrix $\mathcal{A}(G_c)$. We define $\rho = \max\{|\lambda_2(\mathcal{A}(G_c))|, |\lambda_{|V|}(\mathcal{A}(G_c))|\}$ and the spectral gap $\delta(G_c) = (1-\rho) \in (0, 1]$ [9]. With [9], we obtain the following relation between global iterate and spectral gap.

Proposition 2. In [9], Proposition 4 shows the relation between global iterate and the reciprocal of spectral gap is $\mathcal{G}(G_c) \propto \frac{1}{\delta(G_c)}$, and Proposition 5 further bounds the reciprocal of spectral gap $\frac{1}{\delta(G_c)}$ by $\mathcal{O}(\mathcal{H}(G_c))$, where $\mathcal{H}(G_c)$ is the hitting time of G_c (see Definition 6) with the matrix established by eq. (4). Therefore, we yield the following induction

$$\mathcal{G}(G_c) \propto \frac{1}{\delta(G_c)} = \mathcal{O}(\mathcal{H}(G_c)). \tag{10}$$

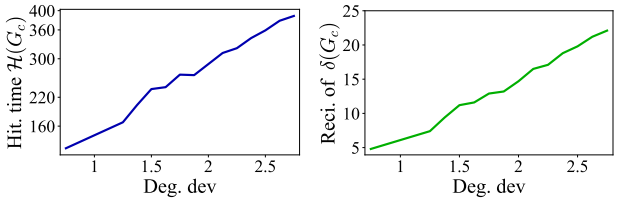
Following Proposition 2, we need to consider the hitting time $\mathcal{H}(G_c)$ (see Definition 6) to predict the global iterate.

Definition 6 (Hitting Time [9])**.** Given a communication matrix $\mathcal{A}(G_c)$ calculated by (4), the entries of relevant hitting time matrix $\mathcal{M}(G_c) \in \mathbb{R}^{|V|\times|V|}$ are defined as

$$m_{ij} = \begin{cases} 0, & \text{if } i = j, \\ 1 + \sum_{k\in V, k\neq j} a_{ik} \cdot m_{kj}, & \text{otherwise}, \end{cases} \tag{11}$$

where m_{ij} is the hitting time (i.e., expected step) from device i to j. The hitting time of communication topology G_c is the *largest* entry in $\mathcal{M}(G_c)$, i.e., $\mathcal{H}(G_c) = \max_{i,j\in V} m_{ij}$.

Remark that the hitting time between i and j is *bidirectional* and the rationale behind $\mathcal{H}(G_c)$ is detailed in Example 2.



(a) Hit. time vs deg. dev. (b) Reci. of spe. gap vs deg. dev.

Fig. 3. (a) and (b) show the relations between the degree deviation and hitting time and the reciprocal of spectral gap, respectively.

Example 2. This example shows the calculation of hitting time with the network in Fig. 2(a), where matrix $\mathcal{M}$ is 6×6. Take the topology in Fig. 2(b) for example. By eq. (11), $m_{AA} = 0$, $m_{BA} = 1 + \frac{2}{3}m_{BA} + \frac{1}{6}m_{CA}$, $m_{CA} = 1 + \frac{1}{6}m_{BA} + \frac{13}{24}m_{CA} + \frac{1}{8}m_{DA} + \frac{1}{6}m_{EA}$, and the rest are omitted. Thus, there are 36 variables attained from 36 equations, and the hitting time is 26, where complete $\mathcal{M}$ is shown in Table II. ■

To clearly present the notion of hitting time (i.e., $\mathcal{H}(G_c)$) and spectral gap (i.e., $\delta(G_c)$) and the relation between them, we depict in Fig. 3 the growth of hitting time and *the reciprocal of spectral gap* when $|V| = 16$ and $|E_c| = 32$ (i.e., each node has 4 neighbors in average). We randomly created 10^4 communication topologies with *fixed number of links* (i.e., 32 links) and observed the degree deviation measured by averaging the *mean absolute deviation* between 4 and current degree of each device. It is obvious to see when the structure of communication topology becomes more *irregular* (i.e., the degree deviation is larger), $\mathcal{H}(G_c)$ and $\frac{1}{\delta(G_c)}$ grow accordingly, which explicitly conforms to the induction in Proposition 2.

Then, we derive the following theorem to show the impact posed by the ratio of DP devices and spectral gap on the convergence rate. Let $X_t = \left[x_0^t\ x_1^t\ \cdots\ x_{|V|-1}^t\right], W_t = \left[w_0^t\ w_1^t\ \cdots\ w_{|V|-1}^t\right], \mathbf{1}_{|V|} \in \mathbb{R}^{|V|}$ denote the concatenation of all local parameters, noises at round t by matrix, and the column vector with each entry equal to 1, respectively. Also, Let $\overline{Y}_t = \frac{(X_t+W_t)\mathbf{1}_{|V|}}{|V|}$, $\overline{X}_t = \frac{X_t\mathbf{1}_{|V|}}{|V|}$, and $\overline{W}_t = \frac{W_t\mathbf{1}_{|V|}}{|V|}$.

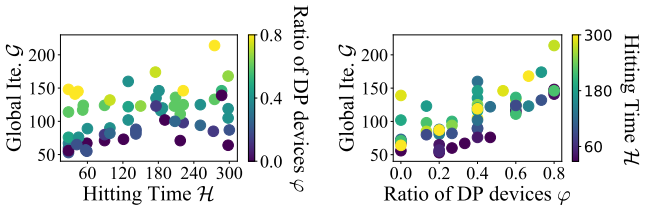
Theorem 1. Suppose each device executes DeepPrivacy T rounds. Let $\varphi \in (0, 1]$ be the ratio of DP devices. The convergence rate of DeepPrivacy satisfies

$$\frac{\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\overline{Y}_t)\|^2}{T} \leq \mathcal{O}\left(\frac{\sum_{t=0}^{T-1}\mathbb{E}\left[f(\overline{Y}_t) - f(\overline{X}_{t+1})\right]}{T\delta(G_c)}\right) + \mathcal{O}\left(\frac{\varphi\mathcal{U}}{T\delta(G_c)}\right), \tag{12}$$

where $\mathcal{U}$ represents the summation of the variance of noises, as defined in line 8 in Algorithm 1, over T rounds (i.e., $\mathcal{U} = \sum_{t\in T}\sum_{i\in V}(\phi_t^i)^2$, where $(\phi_t^i)^2$ denotes the variance of Gaussian distribution in device i at round t). The communication topology is different, say, G_c', if no device perturbs local parameters with noises and it is expected $\delta(G_c) \geq \delta(G_c')$. Let x^* denote the optimal model parameters defined in eq. (5). Then, at round T, we have

$$\frac{\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\overline{X}_t)\|^2}{T} \leq \mathcal{O}\left(\frac{\mathbb{E}[f(\overline{X}_0) - f(x^*)]}{T\delta(G_c')}\right). \tag{13}$$

To further analyze the gap between ineqs. (12) and (13), we derive the following corollary.



(a) Global iterate vs hitting time (b) Global iterate vs DP ratio

Fig. 4. The relation among hitting time $\mathcal{H}(G_c)$, ratio of DP devices φ, and global iterate $\mathcal{G}(G_c)$ at the 68%-accuracy threshold on CIFAR10.

Corollary 1. Let $\alpha = \frac{\delta(G_c)}{\delta(G_c')} \geq 1$ denote the factor that presents the difference between the spectral gap of ineq. (12) and that of ineq. (13). Following the bounds obtained in Theorem 1, the following inequality holds

$$\frac{\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla f(\overline{Y}_t)\|^2 - \|\nabla f(\overline{X}_t)\|^2]}{T} \leq \mathcal{O}\left(\frac{\sum_{t=1}^{T-1}\|\overline{W}_t\|^2 + \varphi\mathcal{U} + f(\overline{Y}_0) - \alpha\|x^*\|^2}{T\delta(G_c)}\right) \tag{14}$$

If $\delta(G_c) = \delta(G_c')$, then $\alpha = 1$.

Due to the page limit, the proofs of Theorem 1 and Corollary 1 are presented in [33]. Theorem 1 shows the relation between perturbed noises and spectral gap. Corollary 1 extends Theorem 1 to deduce that *a suitable amount of imposed noises* can enlarge the spectral gap $\delta(G_c)$ (i.e., reduce the hitting time $\mathcal{H}(G_c)$), thereby increasing the connectivity of communication topologies and accelerating the convergence rate. Specifically, selecting a suitable number of DP devices to generate moderate noises can slightly increase $\overline{W}_t, \varphi, \mathcal{U}$ while greatly increase α in the right-hand side of ineq. (14). Therefore, the three main factors affecting the global iterate $\mathcal{G}(G_c)$ of DeepPrivacy are hitting time $\mathcal{H}(G_c)$, ratio of DP devices φ, and the cumulative variance $\mathcal{U}$. Then, we design a *regression-based* method to predict $\mathcal{G}(G_c)$ based on $\mathcal{H}(G_c)$, φ, and $\mathcal{U}$ in the following.

To select a proper regression function, we refer to the approach in [34] and first fix the variance $\mathcal{U}$ and plot the relation among $\mathcal{G}(G_c)$, $\mathcal{H}(G_c)$, and φ in Fig. 4. We find that the relation of the global iterate for various $\mathcal{H}(G_c)$ and φ is in line with Theorem 1 and Corollary 1. We then adopt the following regression function with a *constant shift* as our fitting function.

$$\mathcal{G}'(G_c) = 0.3165 \cdot \mathcal{H}(G_c) + 121.6365 \cdot \varphi + 18.6251. \tag{15}$$

The configuration of the regression-based method is presented in [33]. We provide the reasons to justify the adoption of the regression-based method to predict global iterate as follows.

- *Fixed privacy budget*: the privacy budget is a critical factor to influence the training performance in DL as shown in Theorem 1. According to the experiments and simulations in [10], it is effective to defend the model-inversion-based attacks by setting the privacy budgets up to a sufficiently small constant (e.g., $\epsilon = 1$, $\delta = 10^{-5}$). With Definition 3 and Proposition 1, the cumulative noises (i.e., the first two terms on the right-hand side of ineq. (14)) are expected to converge to a scalar, which means the degree of deviation caused by perturbed noises can be under control and predicted.
- *Auto hyperparameter optimization*: There exist varying hyperparameters affecting the convergence of machine learning,

such as batch size, learning rate, and so on. Recently, the automated machine learning (AutoML) is an emerging paradigm to *automatically* tune the hyperparameters for a given learning task, the results of which are empirically proved to be highly effective and feasible [35]. In addition, the hyperparameters for DL over the devices are usually the same [5] so there is no need to concern about the influence of the hyperparameters on the convergence.

- *Linear speedup in the number of devices*: Either the loss function is convex or non-convex, the convergence rate of DL is proved to be *linear* in the number of devices [5]. That is, the convergence rate regarding to the varying number of devices can be expected to *grow linearly*, showing regression-based methods are scalable to large-scale cases.

The above three reasons explicitly explain the regression-based method is substantially feasible and scalable to predict global iterate. Therefore, the regression function (i.e., eq. (15)) is used to predict the global iterate to obtain the *pseudo training time* (i.e., predicted global iterate times local iterate) that approximates the physical training time for each candidate solution created in Section V-B. Last, the one with minimum pseudo training time is selected as the solution.

Example 3. Following Examples 1 – 2, the toy example shows the effect of link selection on pseudo training time. Take the topology in Fig. 2(b) for example. The ratio of DP devices φ is 0.5 since devices C, D, E adopt DP to exploit non-social links $\overline{CE}, \overline{DE}$. Recall that the local iterate is 52 (see Example 1) and the hitting time is 26 (see Example 2). Therefore, by eq. (15), $\mathcal{G}'(G_c) \approx 87.7$ and the pseudo training time is 4559.0. ■

B. Construct the Communication Topology

To address the *varying relation between global iterate and DP devices*, it is necessary to examine the effect of every possible ratio of DP devices φ on global iterate. We design an algorithm termed AutoTag. The idea is to construct a candidate solution for each possible number of DP devices and then pick the best one among them as the output. Thus, at most $|V|+1$ candidate solutions, $G_c^0, ..., G_c^{|V|}$, exist. Each candidate G_c^n has n DP devices, where $0 \leq n \leq |V|$. Then, it follows the guide of two scoring methods, social loner score (SLS) and communicative loner score (CLS) (detailed later), to evaluate links in communication topologies under specific conditions to deal with the *trade-off between global and local iterates*. SLS can suggest the suitable loner device (i.e., with fewer neighbors in the social network) to adopt DP so as to *get more bang for the limit of DP devices*. Likewise, CLS provides a measure of proper devices with fewer neighbors, lower local iterate, but higher hitting time. Finally, AutoTag carefully examines the construction progress to find the best topology snapshot.

AutoTag includes the following four phases: 1) Connectivity Guarantee Phase (CGP), 2) Loner Connection Phase (LCP), 3) Network Expanding Phase (NEP), and 4) Snapshot Selection Phase (SSP). Particularly, CGP first constructs an initial solution, where each device has a similar number of neighbors, for each candidate G_c^n. Then, LCP connects the loners (i.e., the devices with low degree in the social network) to increase opportunities for constructing a near-regular communication topology. Afterward, NEP expands each candidate G_c^n by adding the links able to balance the hitting time and local iterate until no link is available. Last, SSP examines each topology snapshot at each iteration and picks the one with the minimum pseudo training time (i.e., the predicted global iterate times the local iterate) as candidate G_c^n. Then, SSP chooses the one with the minimum pseudo training time among all candidate solutions (i.e., $G_c^1, ..., G_c^{|v|}$) as the communication topology G_c. Due to the page limit, the pseudocode is presented in [33].

1) Connectivity Guarantee Phase (CGP): All the involved devices must be connected in G_c^n, where $0 \leq n \leq |V|$. Also, small hitting time usually arises in more regular graphs. Thus, CGP gives the priority to connecting two devices with the lowest degree sum in G_c^n. For tie breaking, it first connects the link included in the social network G_s and with a smaller $(d_{ij} + \max\{r_i, r_j\})$. Meanwhile, two devices should adopt DP to exploit the selected non-social link if the number of DP devices is no greater than n. Otherwise, CGP will discard it. Note that G_c^n is connected after CGP since G_s is connected.

2) Loner Connection Phase (LCP): The loner devices in the social network G_s have fewer links connecting to other devices and thus make G_c^n hard to approximate a regular topology, which is believed to have a lower hitting time compared to the other topologies with the same number of links. Thus, LCP iteratively adds a link (i, j) into G_c^n, where (i, j) has a high hitting time in G_c^n and two low-degree endpoint devices in G_s while leading to a low local iterate. Specifically, LCP iteratively selects the pair of devices with the maximum SLS defined as follows, where $deg_s(i)$ is the degree of device $i \in V$ in G_s.

$$SLS(i,j) = \frac{\max\{m_{ij}, m_{ji}\}}{(d_{ij} + \max\{r_i, r_j\})\ deg_s(i)\ deg_s(j)}. \quad (16)$$

Remark that the selected link may not be in G_s such that DP will be adopted by the two devices if the number of DP devices is not greater than n. Otherwise, the link will be skipped.

3) Network Expanding Phase (NEP): To address the trade-off between global and local iterates, NEP adds the links of devices that tend to have a high hitting time while low-degree endpoint devices in G_c^n and lead to a low local iterate to approximate a near-regular topology. Specifically, NEP iteratively selects the pair with the maximum CLS defined as follows, where $deg_c(i)$ denotes the degree of device $i \in V$ in G_c^n.

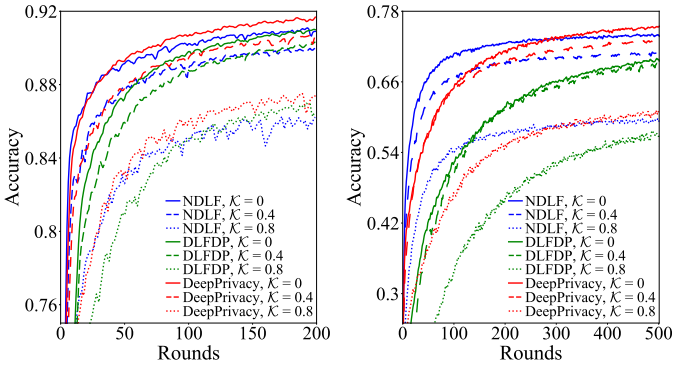
$$CLS(i,j) = \frac{\max\{m_{ij}, m_{ji}\}}{(d_{ij} + \max\{r_i, r_j\})\ deg_c(i)\ deg_c(j)}. \quad (17)$$

4) Snapshot Selection Phase (SSP): For each n, SSP selects the one with the minimum pseudo training time by eqs. (8) and (15) among all the snapshots through all iterations for G_c^n to be candidate G_c^n. Finally, it picks the candidate with the minimum pseudo training time from $G_c^0, ..., G_c^{|V|}$ to be the solution G_c.

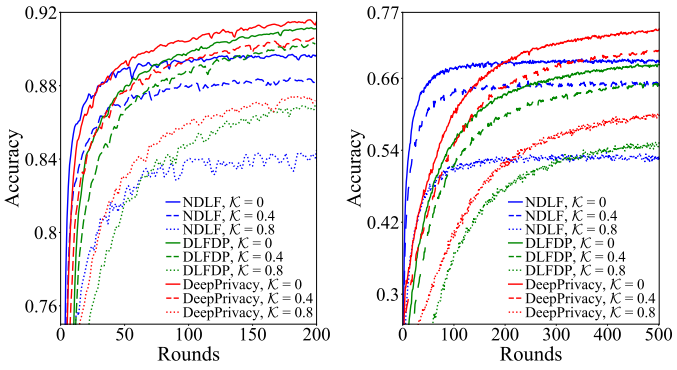
VI. Performance Evaluation

A. Implementation and Simulation Settings

We compare DeepPrivacy with naïve non-convex-based DL framework (NDLF) [14], and the non-convex-based DL framework with full DP (DLFDP) [31]. In particular, NDLF *just* uses the given social network as communication topology (i.e., $E_c = E_s$) for training and exchanging parameters since it does not consider the construction of communication topologies.



(a) $|V|=16$, FMNIST (b) $|V|=16$, CIFAR10



(c) $|V|=32$, FMNIST (d) $|V|=32$, CIFAR10

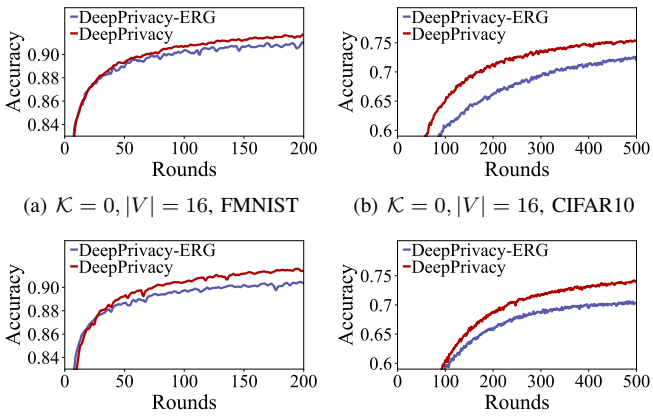
Fig. 5. Performance of three frameworks on FMNIST and CIFAR10 with $|V|=16$ and 32, and $\mathcal{K}=0$, 0.4, and 0.8.

DLFDP can train with *complete* communication topologies (i.e., $E_c = V \times V$) since all devices are appointed to adopt DP mechanism. The experiment setup is detailed as follows.

1) Dataset: For evaluating DeepPrivacy, two well-known benchmarks are adopted, which are CIFAR10 [13] and FMNIST [23]. We evaluate the *top-1* test accuracy on every device separately over the whole dataset and depict the average performance over all devices. For the distribution of social relations and positions of devices, the *real-world* dataset Santander [15], which stores the locations of 16216 SIoT devices in Santander and depicts the relationship (e.g., ownership object relation) among devices, is used to simulate the scenarios. We focus on ownership object relations and static devices for experiments.

2) Simulation Settings: The *computation time* of the SIoT devices is estimated according to the GFLOPS benchmark of *Raspberry Pi Model B series* from RPi2, RPi3, and RPi4, which requires 770s, 312s, and 114s per local round of training, respectively. The *communication time* for transmitting 4-MB model parameters per round depends on the distance between devices. If the distance is less than 100m, the devices can communicate over Wi-Fi and the data rate is at most 72.2 Mbps (802.11n on 2.4 GHz) [36]. If not, the devices communicate over LTE-M since Wi-Fi can only cover some 100m [36]. For LTE-M, two standards are exploited, which are Cat-M1 and Cat-M2. The data rate of LTE-M follows 3GPP releases.

3) Implementation Settings of DL: We implement two magnitudes of devices, which is 16 and 32. The adopted social and physical networks with the same number of devices are extracted from Santander randomly. To implement DL with



(a) $\mathcal{K}=0, |V|=16$, FMNIST (b) $\mathcal{K}=0, |V|=16$, CIFAR10

(c) $\mathcal{K}=0, |V|=32$, FMNIST (d) $\mathcal{K}=0, |V|=32$, CIFAR10

Fig. 6. Performance of two methods to construct communication topologies.

independent-and-identically distributed (IID) and non-IID data, we follow the method of data partitions used in [37]. We control the *skewness* $\mathcal{K} \in [0,1]$ by assigning distinct fractions of non-IID data to each device. For example, if $\mathcal{K}=0.4$, each device is allocated with a data partition, 40% of which belongs to the same class and 60% of which is IID. The input images for training are preprocessed according to [3]. TensorFlow and Keras are used to implement a convolutional neural network (CNN) composed of 2 CLs and 3 FCLs. The details of 2 CLs and 3 FCLs are presented in [33]. The convergence index $\varepsilon = 10^{-5}$, checkpoint round $H=1$, and target round $T=200$, 500 for FMNIST and CIFAR10, respectively. The privacy budget $\epsilon = 1$, $\delta = 10^{-5}$, which follows the suggestions in [10]. Each implementation result is averaged over 10 trials.

B. Performance on Convergence and Accuracy

The results of three frameworks with $|V|=16$ on FMNIST and CIFAR10 are summarized in Figs. 5(a) and 5(b). The convergence rate and accuracy of three frameworks differ from the degree of skewness. The larger the degree of skewness, the worse the performance. The performance of DLFDP is the worst since too much noise deviates the dynamics of DL. NDLF can achieve higher accuracy in less rounds than does DeepPrivacy. However, DeepPrivacy can reach better final accuracy than NDLF since the social links restrict the growth rate of accuracy and DeepPrivacy makes good use of DP to break the limit. The results with $|V|=32$ are summarized in Figs. 5(c) and 5(d). DeepPrivacy still outperforms the others in terms of convergence rate and final accuracy since it strikes balance between noises and hitting time by properly adopting DP. The results in Fig. 5 show that if the balance between the amount of noises and hitting time is well-addressed, training performance can be better, which explicitly proves Corollary 1. To see the interplay between convergence rate and physical time consumption, the results of physical training time with $|V|=16$, 32, $\mathcal{K}=0$ are presented in Table III. DeepPrivacy necessitates far less physical training time than the others regardless of the number of devices for ultimate accuracy threshold (i.e., 91% and 74% for MNIST and CIFAR, respectively). Remark that NDLF and DLFDP cannot achieve some specific accuracy thresholds (e.g., 72%) so there are some empty fields (i.e., notation -) in Table III.

TABLE III
PHYSICAL TRAINING TIME WITH 16 AND 32 DEVICES (HOUR)

Accuracy (FMNIST)	85%	87%	89%	91%
DeepPrivacy-16	2.8 (1x)	**4.1** (1x)	**8.9** (1x)	**26.2** (1x)
NDLF-16	**1.9** (0.7x)	4.5 (1.1x)	10.2 (1.2x)	39.3 (1.5x)
DLFDP-16	6.3 (2.2x)	9.7 (2.3x)	16.6 (1.9x)	42.8 (1.6x)
DeepPrivacy-32	2.8 (1x)	5.2 (1x)	**9.9** (1x)	**27.4** (1x)
NDLF-32	**2.2** (0.8x)	**4.1** (0.8x)	11.5 (1.2x)	-
DLFDP-32	6.6 (2.4x)	10.4 (2x)	18.9 (1.9x)	46.3 (1.7x)
Accuracy (CIFAR10)	65%	68%	72%	74%
DeepPrivacy-16	21.8 (1x)	28.3 (1x)	**48.6** (1x)	**70.2** (1x)
NDLF-16	**8.6** (0.4x)	**13.0** (0.5x)	33.7 (0.7x)	84.9 (1.2x)
DLFDP-16	56.6 (2.6x)	80.4 (2.8x)	-	-
DeepPrivacy-32	31.1 (1x)	40.2 (1x)	**66.8** (1x)	**106.1** (1x)
NDLF-32	**10.3** (0.4x)	**24.0** (0.6x)	-	-
DLFDP-32	66.4 (2.1x)	93.6 (2.3x)	-	-

C. Comparison between AutoTag and the Erdös-Rényi Graph

In numerous DL-based frameworks [31], [38], [39], the *Erdös-Rényi graph* (ERG) is usually considered to be communication topologies since it is proved effective to construct a graph with *high connectivity* [40]. We follow the approach used in [31] to configure DeepPrivacy-ERG where communication topologies are constructed as ERG and two devices are assigned to adopt DP mechanism if the link between them is induced in the constructed ERG but not in the social network. Similarly, we compare DeepPrivacy with DeepPrivacy-ERG on FMNIST and CIFAR10 with $|V| = 16$, 32 and $\mathcal{K} = 0$. *The only difference between the two frameworks is the built-in approach to constructing communication topologies*. The results are summarized in Fig. 6. DeepPrivacy still outperforms DeepPrivacy-ERG regardless of the number of devices since AutoTag in DeepPrivacy can make good use of DP.

VII. CONCLUSION

This paper proposes a DL framework DeepPrivacy engaging partially DP scheme for SIoT scenarios. The framework derives an optimization problem CoTOPO. CoTOPO is very intractable due to the new challenges, i.e., trade-off between global and local iterates, uncertain global iterate, and varying relation between global iterate and DP devices. We propose a novel algorithm AutoTag to subtly make use of two scoring methods to select suitable links and DP devices to substantially reduce physical training time. The experiment results manifest that DeepPrivacy and AutoTag combined outperform the state of the art by more than 20%.

REFERENCES

[1] C.-H. Wang *et al.*, "Collaborative social internet of things in mobile edge networks," *IEEE Int. of Things J.*, vol. 7, no. 12, pp. 11 473–11 491, 2020.

[2] L. Atzori *et al.*, "The social internet of things (SIoT)–when social networks meet the internet of things: Concept, architecture and network characterization," *Comput. Netw.*, vol. 56, no. 16, pp. 3594–3608, 2012.

[3] H. B. McMahan *et al.*, "Communication-efficient learning of deep networks from decentralized data," in *PMLR AISTATS*, 2017.

[4] T. Li *et al.*, "Federated learning: challenges, methods, and future directions," *IEEE Sig. Proc. Mag.*, vol. 37, no. 3, pp. 50–60, 2020.

[5] A. Koloskova, T. Lin, S. U. Stich, and M. Jaggi, "Decentralized deep learning with arbitrary communication compression," in *ICLR*, 2020.

[6] O. Briante *et al.*, "A social and pervasive IoT platform for developing smart environments," in *The Int. of Things for Smart Urban Ecosys.* Springer, 2019, pp. 1–23.

[7] B. Afzal *et al.*, "Enabling IoT platforms for social iot applications: vision, feature mapping, and challenges," *Future Gen. Comp. Sys.*, vol. 92, pp. 718–731, 2019.

[8] S. Shi, X. Chu, and B. Li, "MG-WFBP: Efficient data communication for distributed synchronous SGD algorithms," in *IEEE INFOCOM*, 2019.

[9] A. Nedić, A. Olshevsky, and M. G. Rabbat, "Network topology and communication-computation tradeoffs in decentralized optimization," *Proc. of the IEEE*, vol. 106, no. 5, pp. 953–976, 2018.

[10] B. Hitaj *et al.*, "Deep models under the GAN: information leakage from collaborative deep learning," in *ACM CCS*, 2017.

[11] M. Abadi *et al.*, "Deep learning with differential privacy," in *ACM CCS*, 2016.

[12] C. Li *et al.*, "Differentially private distributed online learning," *IEEE Trans. on Know. and Data Eng.*, vol. 30, pp. 1440–1453, 2018.

[13] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009.

[14] X. Lian *et al.*, "Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent," in *NeurIPS*, 2017.

[15] C. Marche *et al.*, "How to exploit the social Internet of Things: Query generation model and device profiles' dataset," *Comput. Netw.*, vol. 174, p. 107248, 2020.

[16] A. G. Howard *et al.*, "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv:1704.04861*, 2017.

[17] F. N. Iandola *et al.*, "SqueezeNet: Alexnet-level accuracy with 50x fewer parameters and $<$ 0.5 MB model size," *arXiv:1602.07360*, 2016.

[18] P. Stock *et al.*, "And the Bit Goes Down: Revisiting the quantization of neural networks," in *ICLR*, 2020.

[19] R. Ratasuk, N. Mangalvedhe, D. Bhatoolaul, and A. Ghosh, "LTE-M evolution towards 5G massive mtc," in *IEEE Globecom Workshops*, 2017.

[20] L. Lovász *et al.*, "Random walks on graphs: A survey," *Combinatorics, Paul erdos is eighty*, vol. 2, no. 1, pp. 1–46, 1993.

[21] S. Lee and S. Nirjon, "Subflow: A dynamic induced-subgraph strategy toward real-time dnn inference and training," in *IEEE RTAS*, 2020.

[22] M. Lauridsen *et al.*, "Coverage and capacity analysis of LTE-M and NB-IoT in a rural area," in *IEEE VTC-Fall*, 2016.

[23] H. Xiao *et al.*, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv:1708.07747*, 2017.

[24] J. Konečnỳ *et al.*, "Federated learning: Strategies for improving communication efficiency," in *NeurIPS Workshop*, 2016.

[25] S. Wang *et al.*, "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. on Sel. Areas in Comm.*, vol. 37, no. 6, pp. 1205–1221, 2019.

[26] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *IEEE ICC*, 2019.

[27] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. Autom. Control*, vol. 31, pp. 803–812, 1986.

[28] Y. Li *et al.*, "Pipe-SGD: A decentralized pipelined SGD framework for distributed deep net training," in *NeurIPS*, 2018.

[29] K. Wei *et al.*, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Trans. on Info. For. and Sec.*, vol. 15, pp. 3454–3469, 2020.

[30] P. C. Mahawaga Arachchige *et al.*, "Local differential privacy for deep learning," *IEEE Int. of Things J.*, vol. 7, pp. 5827–5842, 2020.

[31] X. Zhang *et al.*, "Private and communication-efficient edge learning: a sparse differential gaussian-masking distributed SGD approach," in *ACM Mobihoc*, 2020.

[32] N. Agarwal *et al.*, "cpSGD: Communication-efficient and differentially-private distributed SGD," in *NeurIPS*, 2018.

[33] C.-W. Ching, H.-S. Huang, C.-A. Yang, Y.-C. Liu, and J.-J. Kuo, "Efficient communication topology via partially differential privacy for decentralized learning (technical report)," Mar 2021. [Online]. Available: https://github.com/lab401b/DL-ICCCN2021-Appendix

[34] H. Zhang *et al.*, "AutoSync: Learning to synchronize for data-parallel distributed deep learning," in *NeurIPS*, 2020.

[35] X. He, K. Zhao, and X. Chu, "Automl: A survey of the state-of-the-art," *Knowledge-Based Systems*, vol. 212, p. 106622, 2021.

[36] S. R. Pokhrel *et al.*, "Adaptive admission control for iot applications in home WiFi networks," *IEEE Trans. on Mob. Comp.*, vol. 19, no. 12, pp. 2731–2742, 2019.

[37] K. Hsieh, A. Phanishayee, O. Mutlu, and P. Gibbons, "The non-IID data quagmire of decentralized machine learning," in *PMLR ICML*, 2020.

[38] A. Reisizadeh, A. Mokhtari, H. Hassani, and R. Pedarsani, "An exact quantized decentralized gradient descent algorithm," *IEEE Trans. on Sig. Proc.*, vol. 67, no. 19, pp. 4934–4947, 2019.

[39] A. Koloskova *et al.*, "Decentralized stochastic optimization and gossip algorithms with compressed communication," in *PMLR ICML*, 2019.

[40] P. Erdős and A. Rényi, "On the evolution of random graphs," *Publ. Math. Inst. Hung. Acad. Sci*, vol. 5, no. 1, pp. 17–60, 1960.

APPENDIX

Before getting down to the theoretical proofs, we define the notations as follows:

- $\|\cdot\|$ denotes l^2-norm.
- $\|\cdot\|_F$ denotes the matrix Frobenius norm.
- $\nabla g(\cdot)$ denotes the gradient of a function g.
- $\mathbf{1}_{|V|}$ denotes the column vector in $\mathbb{R}^n$ with 1 for all elements.
- x^* denotes the optimal solution of $\min_{x\in\mathbb{R}^d} f(x) := \mathbb{E}_{\xi\sim\mathcal{D}}F(x;\xi)$.
- $\lambda_i(\cdot)$ denotes the i-th largest eigenvalue of a matrix.

Also, we provide some practical assumptions on the dynamics of SGD and the local parameters as follows.

Assumption 1. *(Bounded Variance). The variance of the stochastic gradients is bounded on each agent:*

$$\mathbb{E}_{\xi_i}\|\nabla F_i(x,\xi) - \nabla f_i(x)\|^2 \le \sigma^2, \qquad \forall i, x, \tag{18}$$

$$\mathbb{E}_{i}\|\nabla f_i(x) - \nabla f(x)\|^2 \le \varsigma^2, \qquad \forall i, x, \tag{19}$$

$$\mathbb{E}_{\xi_i}\|\nabla F_i(x,\xi)\|^2 \le G^2, \qquad \forall i, x. \tag{20}$$

Assumption 2. *(L-Lipschitzian Gradients). Each function $f_i : \mathbb{R}^N \to \mathbb{R}, \forall i \in n$ is L-smooth, that is*

$$\|\nabla f_i(y) - \nabla f_i(x)\| \le L\|y - x\|, \qquad \forall x, y \in \mathbb{R}^N, i \in n. \tag{21}$$

Assumption 3. *(Initialize from 0). We assume $X_0 = 0$. This assumption simplifies the proof w.l.o.g.*

A. Proof to Proposition 1

Before proving Proposition 1, we derive the following lemma.

Lemma 1 ([1])**.** For any δ, l_2 sensitivity bound Δ, and ϕ such that $\phi \ge \Delta\sqrt{2\log 1.25/\delta}$, the Gaussian mechanism $M^{\phi}(f(D)) := f(D) + Z$, where $Z \sim |\mathcal{V}|(0, \phi^2\mathbf{I}_d)$, is $(\frac{\Delta}{\phi}\sqrt{2\log 1.25/\delta}, \delta)$ differentially private,

where $\mathbf{I}_d$ denotes the identity matrix with dimension equal to d. The proof to Lemma 1 can be found in [2].
Now, we are ready to bound the sensitivity as follows.

Lemma 2. Suppose that Assumption 1 holds. Then, the following inequality holds

$$\Delta_t \le 2\eta_t G \tag{22}$$

The proof can be found in [3]–[5].

Combining Lemmas 1 and 2, DeepPrivacy is (ϵ, δ)-differentially private for the devices that adopt DP by setting $\phi_t = \frac{\Delta_t\sqrt{(2\log 1.25)/\delta}}{\epsilon}$ where $\Delta_t = 2\eta_t G$. The proof is completed.

B. Proof to Theorem 1

The objective function is as follows

$$\min_{x\in\mathbb{R}^d} f(x) := \frac{1}{|V|}\sum_{i=1}^{|V|} f_i(x) = \frac{1}{|V|}\sum_{i=1}^{|V|}\mathbb{E}_{\xi\sim\mathcal{D}_i}F(x;\xi). \tag{23}$$

Two helpful lemmas can be stated:

Lemma 3. Let e_i, ρ denote the i^{th} column and the second largest singular value in $\mathcal{A}$. Under assumption of communication matrix we have

$$\|\frac{\mathbf{1}_{|V|}}{|V|} - \mathcal{A}^k e_i\| \le \rho^k, \forall i \in \{1, 2, ..., |V|\}, k \in \mathbb{Z}^+ \cup \{0\},$$

where $\mathcal{A}^0 = \mathbf{I}$.

The proof to Lemma 3 can be found in [6].

Lemma 4. We have the following inequality under Assumption 1:

$$\mathbb{E}\|\partial f(X_j+W_j)\|^2 \le \sum_{h=1}^{|V|} 3\mathbb{E}L^2\left[\left\|\frac{\sum_{i'=1}^{|V|}x_{j,i'}}{|V|}-x_{j,h}\right\|^2+\left\|\frac{\sum_{i'=1}^{|V|}w_{j,i'}}{|V|}-w_{j,h}\right\|^2\right] + 3|V|\varsigma^2+3\mathbb{E}\left\|\nabla f\left(\frac{(X_j+W_j)\mathbf{1}_{|V|}}{|V|}\right)\mathbf{1}_{|V|}^\top\right\|^2, \forall j.$$

Proof. We consider the upper bound of $\mathbb{E}\|\partial f(X_j+W_j)\|^2$ in the following:

$$\begin{aligned}
&\mathbb{E}\|\partial f(X_j+W_j)\|^2\\
&\le 3\mathbb{E}\left\|\partial f(X_j+W_j)-\partial f\left(\frac{(X_j+W_j)\mathbf{1}_{|V|}}{|V|}\mathbf{1}_{|V|}^\top\right)\right\|^2+3\mathbb{E}\left\|\partial f\left(\frac{(X_j+W_j)\mathbf{1}_{|V|}}{|V|}\mathbf{1}_{|V|}^\top\right)-\nabla f\left(\frac{(X_j+W_j)\mathbf{1}_{|V|}}{|V|}\right)\mathbf{1}_{|V|}^\top\right\|^2\\
&\quad+3\mathbb{E}\left\|\nabla f\left(\frac{(X_j+W_j)\mathbf{1}_{|V|}}{|V|}\right)\mathbf{1}_{|V|}^\top\right\|^2\\
&\overset{(a)}{\le} 3\mathbb{E}\left\|\partial f(X_j+W_j)-\partial f\left(\frac{(X_j+W_j)\mathbf{1}_{|V|}}{|V|}\mathbf{1}_{|V|}^\top\right)\right\|_F^2+3|V|\varsigma^2+3\mathbb{E}\left\|\nabla f\left(\frac{(X_j+W_j)\mathbf{1}_{|V|}}{|V|}\right)\mathbf{1}_{|V|}^\top\right\|^2\\
&\overset{(b)}{\le}\sum_{h=1}^{|V|}3\mathbb{E}L^2\left\|\frac{\sum_{i'=1}^{|V|}x_{j,i'}+w_{j,i'}}{|V|}-(x_{j,h}+w_{j,h})\right\|^2+3|V|\varsigma^2+3\mathbb{E}\left\|\nabla f\left(\frac{(X_j+W_j)\mathbf{1}_{|V|}}{|V|}\right)\mathbf{1}_{|V|}^\top\right\|^2\\
&=\sum_{h=1}^{|V|}3\mathbb{E}L^2\left\|\frac{\sum_{i'=1}^{|V|}x_{j,i'}}{|V|}-x_{j,h}+\frac{\sum_{i'=1}^{|V|}w_{j,i'}}{|V|}-w_{j,h}\right\|^2+3|V|\varsigma^2+3\mathbb{E}\left\|\nabla f\left(\frac{(X_j+W_j)\mathbf{1}_{|V|}}{|V|}\right)\mathbf{1}_{|V|}^\top\right\|^2\\
&=\sum_{h=1}^{|V|}3\mathbb{E}L^2\left[\left\|\frac{\sum_{i'=1}^{|V|}x_{j,i'}}{|V|}-x_{j,h}\right\|^2+\left\|\frac{\sum_{i'=1}^{|V|}w_{j,i'}}{|V|}-w_{j,h}\right\|^2+2\left\langle\frac{\sum_{i'=1}^{|V|}x_{j,i'}}{|V|}-x_{j,h},\frac{\sum_{i'=1}^{|V|}w_{j,i'}}{|V|}-w_{j,h}\right\rangle\right]\\
&\quad+3|V|\varsigma^2+3\mathbb{E}\left\|\nabla f\left(\frac{(X_j+W_j)\mathbf{1}_{|V|}}{|V|}\right)\mathbf{1}_{|V|}^\top\right\|^2\\
&=\sum_{h=1}^{|V|}3\mathbb{E}L^2\left[\left\|\frac{\sum_{i'=1}^{|V|}x_{j,i'}}{|V|}-x_{j,h}\right\|^2+\left\|\frac{\sum_{i'=1}^{|V|}w_{j,i'}}{|V|}-w_{j,h}\right\|^2+2\left\langle\frac{\sum_{i'=1}^{|V|}x_{j,i'}}{|V|}-x_{j,h},\mathbb{E}\left[\frac{\sum_{i'=1}^{|V|}w_{j,i'}}{|V|}-w_{j,h}\right]\right\rangle\right]\\
&\quad+3|V|\varsigma^2+3\mathbb{E}\left\|\nabla f\left(\frac{(X_j+W_j)\mathbf{1}_{|V|}}{|V|}\right)\mathbf{1}_{|V|}^\top\right\|^2\\
&\overset{(c)}{=}\sum_{h=1}^{|V|}3\mathbb{E}L^2\left[\left\|\frac{\sum_{i'=1}^{|V|}x_{j,i'}}{|V|}-x_{j,h}\right\|^2+\left\|\frac{\sum_{i'=1}^{|V|}w_{j,i'}}{|V|}-w_{j,h}\right\|^2\right]+3|V|\varsigma^2+3\mathbb{E}\left\|\nabla f\left(\frac{(X_j+W_j)\mathbf{1}_{|V|}}{|V|}\right)\mathbf{1}_{|V|}^\top\right\|^2
\end{aligned}$$

where (a) and (b) follow from Assumption 1 and Assumption 2, respectively and (c) comes from the fact that noises are drawn from Gaussian distribution with mean equal to 0. This completes the proof. □

We start from $f\left(\frac{X_{t+1}\mathbf{1}_{|V|}}{|V|}\right)$:

$$\begin{aligned}
&\mathbb{E}f\left(\frac{X_{t+1}\mathbf{1}_{|V|}}{|V|}\right)\\
&=\mathbb{E}f\left(\frac{(X_t+W_t)\mathcal{A}\mathbf{1}_{|V|}}{|V|}-\gamma\frac{\partial F\big((X_t+W_t);\xi_t\big)\mathbf{1}_{|V|}}{|V|}\right)\\
&=\mathbb{E}f\left(\frac{(X_t+W_t)\mathbf{1}_{|V|}}{|V|}-\gamma\frac{\partial F\big((X_t+W_t);\xi_t\big)\mathbf{1}_{|V|}}{|V|}\right)\\
&\le\mathbb{E}f\left(\frac{(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\right)-\gamma\mathbb{E}\left\langle\nabla f\left(\frac{(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\right),\frac{\partial f(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\right\rangle+\frac{\gamma^2L}{2}\mathbb{E}\left\|\frac{\sum_{i=1}^{|V|}\nabla F_i\left((x_{t,i}+w_{t,i});\xi_{t,i}\right)}{|V|}\right\|^2.
\end{aligned}\tag{24}$$

Note that for the last term we can split it into two terms:

$$
\begin{aligned}
&\mathbb{E}\left\|\frac{\sum_{i=1}^{|V|}\nabla F_i\Big((x_{t,i}+w_{t,i});\xi_{t,i}\Big)}{|V|}\right\|^2\\
&=\mathbb{E}\left\|\frac{\sum_{i=1}^{|V|}\nabla F_i\Big((x_{t,i}+w_{t,i});\xi_{t,i}\Big)-\sum_{i=1}^{|V|}\nabla f_i(x_{t,i}+w_{t,i})}{|V|}+\frac{\sum_{i=1}^{|V|}\nabla f_i(x_{t,i}+w_{t,i})}{|V|}\right\|^2\\
&=\mathbb{E}\left\|\frac{\sum_{i=1}^{|V|}\nabla F_i\Big((x_{t,i}+w_{t,i});\xi_{t,i}\Big)-\sum_{i=1}^{|V|}\nabla f_i(x_{t,i}+w_{t,i})}{|V|}\right\|^2\\
&\quad+\mathbb{E}\left\|\frac{\sum_{i=1}^{|V|}\nabla f_i(x_{t,i}+w_{t,i})}{|V|}\right\|^2\\
&\quad+2\mathbb{E}\left\langle\frac{\sum_{i=1}^{|V|}\nabla F_i\Big((x_{t,i}+w_{t,i});\xi_{t,i}\Big)-\sum_{i=1}^{|V|}\nabla f_i(x_{t,i}+w_{t,i})}{|V|},\frac{\sum_{i=1}^{|V|}\nabla f_i(x_{t,i}+w_{t,i})}{|V|}\right\rangle\\
&=\mathbb{E}\left\|\frac{\sum_{i=1}^{|V|}\nabla F_i\Big((x_{t,i}+w_{t,i});\xi_{t,i}\Big)-\sum_{i=1}^{|V|}\nabla f_i(x_{t,i}+w_{t,i})}{|V|}\right\|^2\\
&\quad+\mathbb{E}\left\|\frac{\sum_{i=1}^{|V|}\nabla f_i(x_{t,i}+w_{t,i})}{|V|}\right\|^2\\
&\quad+2\mathbb{E}\left\langle\frac{\sum_{i=1}^{|V|}\mathbb{E}_{\xi_{t,i}}\nabla F_i\Big((x_{t,i}+w_{t,i});\xi_{t,i}\Big)-\sum_{i=1}^{|V|}\nabla f_i(x_{t,i}+w_{t,i})}{|V|},\frac{\sum_{i=1}^{|V|}\nabla f_i(x_{t,i}+w_{t,i})}{|V|}\right\rangle\\
&\overset{(a)}{=}\mathbb{E}\left\|\frac{\sum_{i=1}^{|V|}\nabla F_i\Big((x_{t,i}+w_{t,i});\xi_{t,i}\Big)-\sum_{i=1}^{|V|}\nabla f_i(x_{t,i}+w_{t,i})}{|V|}\right\|^2\\
&\quad+\mathbb{E}\left\|\frac{\sum_{i=1}^{|V|}\nabla f_i(x_{t,i}+w_{t,i})}{|V|}\right\|^2,
\end{aligned}
\tag{25}
$$

where (a) holds since $\mathbb{E}[\nabla F_i(\cdot;\cdot)]=\nabla f_i(\cdot),\forall i$. Combining (24) and (25), we can have

$$
\begin{aligned}
&\mathbb{E}f\left(\frac{X_{t+1}\mathbf{1}_{|V|}}{|V|}\right)\\
&\le\mathbb{E}f\left(\frac{(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\right)-\gamma\mathbb{E}\left\langle\nabla f\left(\frac{(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\right),\frac{\partial f(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\right\rangle\\
&\quad+\frac{\gamma^2L}{2}\mathbb{E}\left\|\frac{\sum_{i=1}^{|V|}\nabla F_i\Big((x_{t,i}+w_{t,i});\xi_{t,i}\Big)-\sum_{i=1}^{|V|}\nabla f_i(x_{t,i}+w_{t,i})}{|V|}\right\|^2\\
&\quad+\frac{\gamma^2L}{2}\mathbb{E}\left\|\frac{\sum_{i=1}^{|V|}\nabla f_i(x_{t,i}+w_{t,i})}{|V|}\right\|^2.
\end{aligned}
\tag{26}
$$

The second last term of (26) can be bound by σ as follow:

$$\begin{aligned}
&\frac{\gamma^2 L}{2}\mathbb{E}\left\|\frac{\sum_{i=1}^{|V|}\nabla F_i\Big((x_{t,i}+w_{t,i});\xi_{t,i}\Big)-\sum_{i=1}^{|V|}\nabla f_i(x_{t,i}+w_{t,i})}{|V|}\right\|^2\\
&=\frac{\gamma^2 L}{2|V|^2}\mathbb{E}\left\|\sum_{i=1}^{|V|}\nabla F_i\Big((x_{t,i}+w_{t,i});\xi_{t,i}\Big)-\sum_{i=1}^{|V|}\nabla f_i(x_{t,i}+w_{t,i})\right\|^2\\
&=\frac{\gamma^2 L}{2|V|^2}\mathbb{E}\left\|\sum_{i=1}^{|V|}\Big[\nabla F_i\Big((x_{t,i}+w_{t,i});\xi_{t,i}\Big)-\nabla f_i(x_{t,i}+w_{t,i})\Big]\right\|^2\\
&\le\frac{\gamma^2 L}{2|V|^2}\sum_{i=1}^{|V|}\mathbb{E}\left\|\nabla F_i\Big((x_{t,i}+w_{t,i});\xi_{t,i}\Big)-\nabla f_i(x_{t,i}+w_{t,i})\right\|^2\\
&\overset{(a)}{\le}\frac{\gamma^2 L}{2|V|^2}|V|\cdot\sigma^2\\
&=\frac{\gamma^2 L}{2|V|}\sigma^2,
\end{aligned}\tag{27}$$

where (a) follows from Assumption 1. Thus (26) can combine the bound of (27) and we have

$$\begin{aligned}
&\mathbb{E}f\Big(\frac{X_{t+1}\mathbf{1}_{|V|}}{|V|}\Big)\\
&\le\mathbb{E}f\Big(\frac{(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\Big)-\gamma\mathbb{E}\Big\langle\nabla f\Big(\frac{(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\Big),\frac{\partial f(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\Big\rangle+\frac{\gamma^2 L\sigma^2}{2|V|}\\
&\quad+\frac{\gamma^2 L}{2}\mathbb{E}\left\|\frac{\sum_{i=1}^{|V|}\nabla f_i(x_{t,i}+w_{t,i})}{|V|}\right\|^2.\\
&\overset{(a)}{=}\mathbb{E}f\Big(\frac{(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\Big)-\frac{\gamma-\gamma^2 L}{2}\mathbb{E}\left\|\frac{\partial f(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\right\|^2-\frac{\gamma}{2}\mathbb{E}\left\|\nabla f\Big(\frac{(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\Big)\right\|^2+\frac{\gamma^2 L\sigma^2}{2|V|}\\
&\quad+\frac{\gamma}{2}\mathbb{E}\left\|\nabla f\Big(\frac{(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\Big)-\frac{\partial f(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\right\|^2,
\end{aligned}\tag{28}$$

where (a) comes from the fact that $2\langle a,b\rangle=\|a\|^2+\|b\|^2-\|a-b\|^2$. The last term of (28) can be bound as follow:

$$\begin{aligned}
&\mathbb{E}\left\|\nabla f\Big(\frac{(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\Big)-\frac{\partial f(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\right\|^2\\
&=\mathbb{E}\left\|\nabla f\Big(\frac{\sum_{i'=1}^{|V|}x_{t,i'}+w_{t,i'}}{|V|}\Big)-\sum_{i=1}^{|V|}\frac{\nabla f_i(x_{t,i}+w_{t,i})}{|V|}\right\|^2\\
&\overset{(a)}{=}\mathbb{E}\left\|\frac{1}{|V|}\sum_{i=1}^{|V|}\nabla f_i\Big(\frac{\sum_{i'=1}^{|V|}x_{t,i'}+w_{t,i'}}{|V|}\Big)-\sum_{i=1}^{|V|}\frac{\nabla f_i(x_{t,i}+w_{t,i})}{|V|}\right\|^2\\
&\le\frac{1}{|V|}\sum_{i=1}^{|V|}\mathbb{E}\left\|\nabla f_i\Big(\frac{\sum_{i'=1}^{|V|}x_{t,i'}+w_{t,i'}}{|V|}\Big)-\nabla f_i(x_{t,i}+w_{t,i})\right\|^2\\
&\overset{(b)}{=}\frac{L^2}{|V|}\sum_{i=1}^{|V|}\mathbb{E}\left\|\Big(\frac{\sum_{i'=1}^{|V|}x_{t,i'}+w_{t,i'}}{|V|}\Big)-(x_{t,i}+w_{t,i})\right\|^2,
\end{aligned}\tag{29}$$

where (a) follows from the objective (23) and (b) holds due to Assumption 2. Eq. (29) can be seen as the l^2 distance of the local parameters on the i-th node from the averaged local parameters on all nodes. Then we can bound (28) by bounding (29) so we can have:

$$\mathcal{Q}_{t,i}\coloneqq\mathbb{E}\left\|\Big(\frac{\sum_{i'=1}^{|V|}x_{t,i'}+w_{t,i'}}{|V|}\Big)-(x_{t,i}+w_{t,i})\right\|^2$$

$$
\begin{aligned}
&= \mathbb{E}\left\| \left(\frac{(X_t + W_t)\mathbf{1}_{|V|}}{|V|} \right) - (X_t + W_t)e_i \right\|^2 \\
&= \mathbb{E}\left\| \frac{X_t \mathbf{1}_{|V|}}{|V|} + \frac{W_t \mathbf{1}_{|V|}}{|V|} - (X_t e_i + W_t e_i) \right\|^2 \\
&= \mathbb{E}\left\| \frac{(X_{t-1} + W_{t-1})\mathcal{A}\mathbf{1}_{|V|} - \gamma \partial F\Big((X_{t-1} + W_{t-1}); \xi_{t-1}\Big)\mathbf{1}_{|V|}}{|V|} + \frac{W_t \mathbf{1}_{|V|}}{|V|} \right. \\
&\quad \left. - \left[(X_{t-1} + W_{t-1})\mathcal{A}e_i - \gamma \partial F\Big((X_{t-1} + W_{t-1}); \xi_{t-1}\Big)e_i + W_t e_i \right] \right\|^2 \\
&= \mathbb{E}\left\| \frac{(X_{t-1} + W_{t-1})\mathbf{1}_{|V|} - \gamma \partial F\Big((X_{t-1} + W_{t-1}); \xi_{t-1}\Big)\mathbf{1}_{|V|}}{|V|} + \frac{W_t \mathbf{1}_{|V|}}{|V|} \right. \\
&\quad \left. - \left[(X_{t-1} + W_{t-1})\mathcal{A}e_i - \gamma \partial F\Big((X_{t-1} + W_{t-1}); \xi_{t-1}\Big)e_i + W_t e_i \right] \right\|^2 \\
&= \mathbb{E}\left\| \frac{X_{t-1}\mathbf{1}_{|V|} - \gamma \partial F\Big((X_{t-1} + W_{t-1}); \xi_{t-1}\Big)\mathbf{1}_{|V|}}{|V|} + \frac{W_{t-1}\mathbf{1}_{|V|}}{|V|} + \frac{W_t \mathbf{1}_{|V|}}{|V|} \right. \\
&\quad \left. - \left[X_{t-1}\mathcal{A}e_i - \gamma \partial F\Big((X_{t-1} + W_{t-1}); \xi_{t-1}\Big)e_i + W_{t-1}\mathcal{A}e_i + W_t e_i \right] \right\|^2 \\
&= \mathbb{E}\left\| \frac{X_0 \mathbf{1}_{|V|} - \sum_{i=0}^{t-1} \gamma \partial F\Big((X_i + W_i); \xi_i\Big)\mathbf{1}_{|V|}}{|V|} + \frac{1}{|V|}\sum_{i=0}^{t-1} W_i \mathbf{1}_{|V|} \right. \\
&\quad \left. - \left[X_0 \mathcal{A}^t e_i - \sum_{j=0}^{t-1} \gamma \partial F\Big((X_j + W_j); \xi_j\Big)\mathcal{A}^{t-j-1}e_i + \sum_{j=0}^{t-1} W_j \mathcal{A}^{t-j} e_i \right] \right\|^2 \\
&= \mathbb{E}\left\| X_0 \left(\frac{\mathbf{1}_{|V|}}{|V|} - \mathcal{A}^t e_i \right) - \sum_{j=0}^{t-1} \gamma \partial F\Big((X_j + W_j); \xi_j\Big)\left(\frac{\mathbf{1}_{|V|}}{|V|} - \mathcal{A}^{t-j-1}e_i \right) + \sum_{j=0}^{t-1} W_j \left(\frac{\mathbf{1}_{|V|}}{|V|} - \mathcal{A}^{t-j}e_i \right) \right\|^2 \\
&\overset{(a)}{=} \mathbb{E}\left\| - \sum_{j=0}^{t-1} \gamma \partial F\Big((X_j + W_j); \xi_j\Big)\left(\frac{\mathbf{1}_{|V|}}{|V|} - \mathcal{A}^{t-j-1}e_i \right) + \sum_{j=0}^{t-1} W_j \left(\frac{\mathbf{1}_{|V|}}{|V|} - \mathcal{A}^{t-j}e_i \right) \right\|^2 \\
&= \mathbb{E}\left\| \sum_{j=0}^{t-1} \gamma \partial F\Big((X_j + W_j); \xi_j\Big)\left(\frac{\mathbf{1}_{|V|}}{|V|} - \mathcal{A}^{t-j-1}e_i \right) \right\|^2 + \mathbb{E}\left\| \sum_{j=0}^{t-1} W_j \left(\frac{\mathbf{1}_{|V|}}{|V|} - \mathcal{A}^{t-j}e_i \right) \right\|^2 \\
&\quad - 2\mathbb{E}\left\langle - \sum_{j=0}^{t-1} \gamma \partial F\Big((X_j + W_j); \xi_j\Big)\left(\frac{\mathbf{1}_{|V|}}{|V|} - \mathcal{A}^{t-j-1}e_i \right), \sum_{j=0}^{t-1} W_j \left(\frac{\mathbf{1}_{|V|}}{|V|} - \mathcal{A}^{t-j}e_i \right) \right\rangle \\
&= \mathbb{E}\left\| \sum_{j=0}^{t-1} \gamma \partial F\Big((X_j + W_j); \xi_j\Big)\left(\frac{\mathbf{1}_{|V|}}{|V|} - \mathcal{A}^{t-j-1}e_i \right) \right\|^2 + \mathbb{E}\left\| \sum_{j=0}^{t-1} W_j \left(\frac{\mathbf{1}_{|V|}}{|V|} - \mathcal{A}^{t-j}e_i \right) \right\|^2 \\
&\quad - 2\mathbb{E}\left\langle - \sum_{j=0}^{t-1} \gamma \partial F\Big((X_j + W_j); \xi_j\Big)\left(\frac{\mathbf{1}_{|V|}}{|V|} - \mathcal{A}^{t-j-1}e_i \right), \mathbb{E}\sum_{j=0}^{t-1} W_j \left(\frac{\mathbf{1}_{|V|}}{|V|} - \mathcal{A}^{t-j}e_i \right) \right\rangle \\
&\overset{(b)}{=} \underbrace{\mathbb{E}\left\| \sum_{j=0}^{t-1} \gamma \partial F\Big((X_j + W_j); \xi_j\Big)\left(\frac{\mathbf{1}_{|V|}}{|V|} - \mathcal{A}^{t-j-1}e_i \right) \right\|^2}_{=:T_1} + \underbrace{\mathbb{E}\left\| \sum_{j=0}^{t-1} W_j \left(\frac{\mathbf{1}_{|V|}}{|V|} - \mathcal{A}^{t-j}e_i \right) \right\|^2}_{=:T_2}
\end{aligned}
$$

where (a) follows from Assumption 3 and (b) comes from the fact that noises are drawn from Gaussian distribution with mean equal to 0. Then we give bound on T_1 and T_2.

$$
T_1 = \gamma^2 \mathbb{E}\left\| \sum_{j=0}^{t-1} \partial F\Big((X_j + W_j); \xi_j\Big)\left(\frac{\mathbf{1}_{|V|}}{|V|} - \mathcal{A}^{t-j-1}e_i \right) \right\|^2
$$

$$
\begin{aligned}
&\leq 2\gamma^2\mathbb{E}\bigg\|\sum_{j=0}^{t-1}\bigg[\partial F\Big((X_j+W_j);\xi_j\Big)-\partial f(X_j+W_j)\bigg]\Big(\frac{\mathbf{1}_{|V|}}{|V|}-\mathcal{A}^{t-j-1}e_i\Big)\bigg\|^2\\
&\quad+2\gamma^2\mathbb{E}\bigg\|\sum_{j=0}^{t-1}\partial f(X_j+W_j)\Big(\frac{\mathbf{1}_{|V|}}{|V|}-\mathcal{A}^{t-j-1}e_i\Big)\bigg\|^2\\
&=2\gamma^2\sum_{j=0}^{t-1}\mathbb{E}\bigg\|\bigg[\partial F\Big((X_j+W_j);\xi_j\Big)-\partial f(X_j+W_j)\bigg]\Big(\frac{\mathbf{1}_{|V|}}{|V|}-\mathcal{A}^{t-j-1}e_i\Big)\bigg\|^2\\
&\quad+2\gamma^2\mathbb{E}\bigg\|\sum_{j=0}^{t-1}\partial f(X_j+W_j)\Big(\frac{\mathbf{1}_{|V|}}{|V|}-\mathcal{A}^{t-j-1}e_i\Big)\bigg\|^2\\
&\leq 2\gamma^2\sum_{j=0}^{t-1}\mathbb{E}\bigg\|\partial F\Big((X_j+W_j);\xi_j\Big)-\partial f(X_j+W_j)\bigg\|^2\bigg\|\frac{\mathbf{1}_{|V|}}{|V|}-\mathcal{A}^{t-j-1}e_i\bigg\|^2\\
&\quad+2\gamma^2\mathbb{E}\bigg\|\sum_{j=0}^{t-1}\partial f(X_j+W_j)\Big(\frac{\mathbf{1}_{|V|}}{|V|}-\mathcal{A}^{t-j-1}e_i\Big)\bigg\|^2\\
&\leq 2\gamma^2|V|\sigma^2\sum_{j=0}^{t-1}\rho^{t-j-1}+2\gamma^2\mathbb{E}\bigg\|\sum_{j=0}^{t-1}\partial f(X_j+W_j)\Big(\frac{\mathbf{1}_{|V|}}{|V|}-\mathcal{A}^{t-j-1}e_i\Big)\bigg\|^2\\
&\leq \frac{2\gamma^2|V|\sigma^2}{1-\rho}+2\gamma^2\mathbb{E}\bigg\|\sum_{j=0}^{t-1}\partial f(X_j+W_j)\Big(\frac{\mathbf{1}_{|V|}}{|V|}-\mathcal{A}^{t-j-1}e_i\Big)\bigg\|^2\\
&\leq \frac{2\gamma^2|V|\sigma^2}{1-\rho}+2\gamma^2\sum_{j=0}^{t-1}\mathbb{E}\bigg\|\partial f(X_j+W_j)\Big(\frac{\mathbf{1}_{|V|}}{|V|}-\mathcal{A}^{t-j-1}e_i\Big)\bigg\|^2\\
&\leq \frac{2\gamma^2|V|\sigma^2}{1-\rho}+2\gamma^2\sum_{j=0}^{t-1}\mathbb{E}\|\partial f(X_j+W_j)\|^2\bigg\|\frac{\mathbf{1}_{|V|}}{|V|}-\mathcal{A}^{t-j-1}e_i\bigg\|^2\\
&\overset{(a)}{\leq} \frac{2\gamma^2|V|\sigma^2}{1-\rho}\\
&\quad+6L^2\gamma^2\sum_{j=0}^{t-1}\sum_{h=1}^{|V|}\mathbb{E}\bigg[\bigg(\bigg\|\frac{\sum_{i'=1}^{|V|}x_{j,i'}}{|V|}-x_{j,h}\bigg\|^2+\bigg\|\frac{\sum_{i'=1}^{|V|}w_{j,i'}}{|V|}-w_{j,h}\bigg\|^2\bigg)\bigg\|\frac{\mathbf{1}_{|V|}}{|V|}-\mathcal{A}^{t-j-1}e_i\bigg\|^2\bigg]\\
&\quad+\frac{6|V|\varsigma^2\gamma^2}{1-\rho}+6\gamma^2\sum_{j=0}^{t-1}\mathbb{E}\bigg\|\nabla f\Big(\frac{(X_j+W_j)\mathbf{1}_{|V|}}{|V|}\Big)\mathbf{1}_{|V|}^\top\bigg\|^2\bigg\|\frac{\mathbf{1}_{|V|}}{|V|}-\mathcal{A}^{t-j-1}e_i\bigg\|^2\\
&\overset{(b)}{\leq} \frac{2\gamma^2|V|\sigma^2}{1-\rho}+\frac{6|V|\varsigma^2\gamma^2}{1-\rho}+6L^2\gamma^2\sum_{j=0}^{t-1}\sum_{h=1}^{|V|}\mathbb{E}\bigg[\bigg(\bigg\|\frac{\sum_{i'=1}^{|V|}x_{j,i'}}{|V|}-x_{j,h}\bigg\|^2+\bigg\|\frac{\sum_{i'=1}^{|V|}w_{j,i'}}{|V|}-w_{j,h}\bigg\|^2\bigg)\rho^{t-j-1}\bigg]\\
&\quad+6\gamma^2\sum_{j=0}^{t-1}\mathbb{E}\bigg\|\nabla f\Big(\frac{(X_j+W_j)\mathbf{1}_{|V|}}{|V|}\Big)\mathbf{1}_{|V|}^\top\bigg\|^2\rho^{t-j-1}
\end{aligned}
\tag{30}
$$

Then T_2 can be bound as follows:

$$
\begin{aligned}
T_2&=\mathbb{E}\bigg\|\sum_{j=0}^{t-1}W_j\Big(\frac{\mathbf{1}_{|V|}}{|V|}-\mathcal{A}^{t-j}e_i\Big)\bigg\|^2\\
&\leq\sum_{j=0}^{t-1}\mathbb{E}\bigg\|W_j\Big(\frac{\mathbf{1}_{|V|}}{|V|}-\mathcal{A}^{t-j}e_i\Big)\bigg\|^2\\
&\leq\sum_{j=0}^{t-1}\mathbb{E}\|W_j\|^2\bigg\|\Big(\frac{\mathbf{1}_{|V|}}{|V|}-\mathcal{A}^{t-j}e_i\Big)\bigg\|^2
\end{aligned}
$$

$$
\begin{aligned}
&\leq \sum_{j=0}^{t-1} \mathbb{E}\|W_j\|^2 \rho^{t-j} \\
&\leq \sum_{j=0}^{t-1} \sum_{i=1}^{d} \sum_{l=1}^{|V|} \mathbb{E}|w_{il}|^2 \rho^{t-j} \\
&\leq \varphi d|V| \sum_{j=0}^{t-1} \phi_j^2 \rho^{t-j}
\end{aligned}
$$

Combining T_1 and T_2, we can obtain

$$
\begin{aligned}
\mathcal{Q}_{t,i} \leq & \frac{2\gamma^2|V|\sigma^2}{1-\rho} + \frac{6|V|\varsigma^2\gamma^2}{1-\rho} \\
& + 6L^2\gamma^2 \sum_{j=0}^{t-1} \sum_{h=1}^{|V|} \mathbb{E}\Bigg[\Bigg(\Bigg\|\frac{\sum_{i'=1}^{|V|} x_{j,i'}}{|V|} - x_{j,h}\Bigg\|^2 + \Bigg\|\frac{\sum_{i'=1}^{|V|} w_{j,i'}}{|V|} - w_{j,h}\Bigg\|^2\Bigg)\rho^{t-j-1}\Bigg] \\
& + 6\gamma^2 \sum_{j=0}^{t-1} \mathbb{E}\Bigg\|\nabla f\Bigg(\frac{(X_j+W_j)\mathbf{1}_{|V|}}{|V|}\Bigg)\mathbf{1}_{|V|}^\top\Bigg\|^2 \rho^{t-j-1} + \varphi d|V| \sum_{j=0}^{t-1} \phi_j^2 \rho^{t-j} \\
\leq & \frac{2\gamma^2|V|\sigma^2}{1-\rho} + \frac{6|V|\varsigma^2\gamma^2}{1-\rho} + 6L^2\gamma^2 \sum_{j=0}^{t-1} \sum_{h=1}^{|V|} \mathbb{E}[\mathcal{Q}_{j,h}\rho^{t-j-1}] \\
& + 6\gamma^2 \sum_{j=0}^{t-1} \mathbb{E}\Bigg\|\nabla f\Bigg(\frac{(X_j+W_j)\mathbf{1}_{|V|}}{|V|}\Bigg)\mathbf{1}_{|V|}^\top\Bigg\|^2 \rho^{t-j-1} + \varphi d|V| \sum_{j=0}^{t-1} \phi_j^2 \rho^{t-j}
\end{aligned} \tag{31}
$$

Now $T_1 + T_2$ is bound and recall that (29) asks for average performance on all nodes, which can be defined by:

$$
\begin{aligned}
\mathbb{E}\mathcal{P}_t := & \frac{\sum_{i=1}^{n} \mathbb{E}\mathcal{Q}_{t,i}}{|V|} \\
\leq & \frac{2\gamma^2|V|\sigma^2}{1-\rho} + \frac{6|V|\varsigma^2\gamma^2}{1-\rho} + 6|V|L^2\gamma^2 \sum_{j=0}^{t-1} \mathbb{E}[\mathcal{P}_j\rho^{t-j-1}] \\
& + 6\gamma^2 \sum_{j=0}^{t-1} \mathbb{E}\Bigg\|\nabla f\Bigg(\frac{(X_j+W_j)\mathbf{1}_{|V|}}{|V|}\Bigg)\mathbf{1}_{|V|}^\top\Bigg\|^2 \rho^{t-j-1} + \varphi d|V| \sum_{j=0}^{t-1} \phi_j^2 \rho^{t-j}
\end{aligned}
$$

Summing from $t = 0$ to $T - 1$ we have:

$$
\begin{aligned}
\sum_{t=0}^{T-1} \mathbb{E}\mathcal{P}_t \leq & \frac{2\gamma^2|V|\sigma^2}{1-\rho}T + \frac{6|V|\varsigma^2\gamma^2}{1-\rho}T + 6|V|L^2\gamma^2 \sum_{t=0}^{T-1} \sum_{j=0}^{t-1} \mathbb{E}[\mathcal{P}_j\rho^{t-j-1}] \\
& + 6\gamma^2 \sum_{t=0}^{T-1} \sum_{j=0}^{t-1} \mathbb{E}\Bigg\|\nabla f\Bigg(\frac{(X_j+W_j)\mathbf{1}_{|V|}}{|V|}\Bigg)\mathbf{1}_{|V|}^\top\Bigg\|^2 \rho^{t-j-1} + \varphi d|V| \sum_{t=0}^{T-1} \sum_{j=0}^{t-1} \phi_j^2 \rho^{t-j} \\
\overset{(a)}{\leq} & \frac{2\gamma^2|V|\sigma^2}{1-\rho}T + \frac{6|V|\varsigma^2\gamma^2}{1-\rho}T + \frac{6|V|L^2\gamma^2}{1-\rho} \sum_{t=0}^{T-1} \mathbb{E}\mathcal{P}_t \\
& + \frac{6\gamma^2}{1-\rho} \sum_{t=0}^{T-1} \mathbb{E}\Bigg\|\nabla f\Bigg(\frac{(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\Bigg)\mathbf{1}_{|V|}^\top\Bigg\|^2 + \frac{\varphi d|V|}{(1-\rho)} \sum_{t=0}^{T-1} \phi_t^2
\end{aligned}
$$

where (a) can be achieved by rearranging the summations. Then by rearranging the terms we can obtain the bound for the summation of $\mathbb{E}\mathcal{P}_t$'s from $t = 0$ to $T - 1$:

$$
\begin{aligned}
\Bigg(1 - \frac{6|V|L^2\gamma^2}{1-\rho}\Bigg) \sum_{t=0}^{T-1} \mathbb{E}\mathcal{P}_t \leq & \frac{2\gamma^2|V|\sigma^2}{1-\rho}T + \frac{6|V|\varsigma^2\gamma^2}{1-\rho}T + \frac{6\gamma^2}{1-\rho} \sum_{t=0}^{T-1} \mathbb{E}\Bigg\|\nabla f\Bigg(\frac{(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\Bigg)\mathbf{1}_{|V|}^\top\Bigg\|^2 \\
& + \frac{6\gamma^2}{1-\rho} \sum_{t=0}^{T-1} \mathbb{E}\Bigg\|\nabla f\Bigg(\frac{(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\Bigg)\mathbf{1}_{|V|}^\top\Bigg\|^2 + \frac{\varphi d|V|}{(1-\rho)} \sum_{t=0}^{T-1} \phi_t^2
\end{aligned}
$$

$$\sum_{t=0}^{T-1}\mathbb{E}\mathcal{P}_t \leq \frac{2\gamma^2|V|\sigma^2 T}{(1-\rho)\Big(1-\frac{6|V|L^2\gamma^2}{1-\rho}\Big)} + \frac{6|V|\varsigma^2\gamma^2 T}{(1-\rho)\Big(1-\frac{6|V|L^2\gamma^2}{1-\rho}\Big)} + \frac{6\gamma^2}{(1-\rho)\Big(1-\frac{6|V|L^2\gamma^2}{1-\rho}\Big)}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f\Big(\frac{(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\Big)\mathbf{1}_{|V|}^\top\right\|^2 + \frac{\varphi d|V|}{(1-\rho)}\sum_{t=0}^{T-1}\phi_t^2 \tag{32}$$

Substituting the last term of (28) with (29) and (32), we have

$$\mathbb{E}f\Big(\frac{X_{t+1}\mathbf{1}_{|V|}}{|V|}\Big) \leq \mathbb{E}f\Big(\frac{(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\Big) - \frac{\gamma-\gamma^2L}{2}\mathbb{E}\left\|\frac{\partial f(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\right\|^2 - \frac{\gamma}{2}\mathbb{E}\left\|\nabla f\Big(\frac{(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\Big)\right\|^2 + \frac{\gamma^2L\sigma^2}{2|V|} + \frac{\gamma}{2}L^2\mathbb{E}\mathcal{P}_t \tag{33}$$

Summing from $t=0$ to $T-1$ we get:

$$\begin{aligned}
&\frac{\gamma-\gamma^2L}{2}\sum_{t=0}^{T-1}\mathbb{E}\left\|\frac{\partial f(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\right\|^2 + \frac{\gamma}{2}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f\Big(\frac{(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\Big)\right\|^2 \\
&\leq \sum_{t=0}^{T-1}\mathbb{E}\left[f\Big(\frac{(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\Big) - f\Big(\frac{X_{t+1}\mathbf{1}_{|V|}}{|V|}\Big)\right] + \frac{\gamma^2TL\sigma^2}{2|V|} + \frac{L^2\gamma^3n\sigma^2T}{(1-\rho)\Big(1-\frac{6|V|L^2\gamma^2}{1-\rho}\Big)} + \frac{3L^2n\varsigma^2\gamma^3T}{(1-\rho)\Big(1-\frac{6|V|L^2\gamma^2}{1-\rho}\Big)} \\
&\quad + \frac{3L^2\gamma^3}{(1-\rho)\Big(1-\frac{6|V|L^2\gamma^2}{1-\rho}\Big)}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f\Big(\frac{(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\Big)\mathbf{1}_{|V|}^\top\right\|^2 + \frac{\gamma L^2\varphi d|V|\sum_{t=0}^{T-1}\phi_t^2}{2(1-\rho)\Big(1-\frac{6|V|L^2\gamma^2}{1-\rho}\Big)} \\
&= \sum_{t=0}^{T-1}\mathbb{E}\left[f\Big(\frac{(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\Big) - f\Big(\frac{X_{t+1}\mathbf{1}_{|V|}}{|V|}\Big)\right] + \frac{\gamma^2TL\sigma^2}{2|V|} + \frac{L^2\gamma^3n\sigma^2T}{(1-\rho)\Big(1-\frac{6|V|L^2\gamma^2}{1-\rho}\Big)} + \frac{3L^2n\varsigma^2\gamma^3T}{(1-\rho)\Big(1-\frac{6|V|L^2\gamma^2}{1-\rho}\Big)} \\
&\quad + \frac{3|V|L^2\gamma^3}{(1-\rho)\Big(1-\frac{6|V|L^2\gamma^2}{1-\rho}\Big)}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f\Big(\frac{(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\Big)\right\|^2 + \frac{\gamma L^2\varphi d|V|\sum_{t=0}^{T-1}\phi_t^2}{2(1-\rho)\Big(1-\frac{6|V|L^2\gamma^2}{1-\rho}\Big)},
\end{aligned}$$

By rearranging the inequality above, we obtain:

$$\begin{aligned}
&\frac{\gamma-\gamma^2L}{2}\sum_{t=0}^{T-1}\mathbb{E}\left\|\frac{\partial f(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\right\|^2 + \Big(\frac{\gamma}{2} - \frac{3|V|L^2\gamma^3}{(1-\rho)\Big(1-\frac{6|V|L^2\gamma^2}{1-\rho}\Big)}\Big)\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f\Big(\frac{(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\Big)\right\|^2 \\
&\leq \sum_{t=0}^{T-1}\mathbb{E}\left[f\Big(\frac{(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\Big) - f\Big(\frac{X_{t+1}\mathbf{1}_{|V|}}{|V|}\Big)\right] + \frac{\gamma^2TL\sigma^2}{2|V|} + \frac{L^2\gamma^3|V|T(\sigma^2+3\varsigma^2)}{(1-\rho)\Big(1-\frac{6|V|L^2\gamma^2}{1-\rho}\Big)} \\
&\quad + \frac{\gamma L^2\varphi d|V|\sum_{t=0}^{T-1}\phi_t^2}{2(1-\rho)\Big(1-\frac{6|V|L^2\gamma^2}{1-\rho}\Big)}
\end{aligned} \tag{34}$$

Multiplying the left and the right side of (34) by $\frac{\gamma}{T}$, we obtain:

$$\begin{aligned}
&\frac{1-\gamma L}{2T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\frac{\partial f(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\right\|^2 + \frac{1}{T}\Big(\frac{1}{2} - \frac{3|V|L^2\gamma^2}{(1-\rho)\Big(1-\frac{6|V|L^2\gamma^2}{1-\rho}\Big)}\Big)\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f\Big(\frac{(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\Big)\right\|^2 \\
&\leq \frac{\sum_{t=0}^{T-1}\mathbb{E}\left[f\Big(\frac{(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\Big) - f\Big(\frac{X_{t+1}\mathbf{1}_{|V|}}{|V|}\Big)\right]}{\gamma T} + \frac{\gamma L\sigma^2}{2|V|} + \frac{L^2\gamma^2|V|(\sigma^2+3\varsigma^2)}{(1-\rho)\Big(1-\frac{6|V|L^2\gamma^2}{1-\rho}\Big)} + \frac{L^2\varphi d|V|\sum_{t=0}^{T-1}\phi_t^2}{2T(1-\rho)\Big(1-\frac{6|V|L^2\gamma^2}{1-\rho}\Big)}
\end{aligned} \tag{35}$$

Let $\gamma \leq \frac{1}{L}$ and remove the $\left\|\frac{\partial f(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\right\|^2$ terms on the left hand side of (35) , we obtain:

$$\begin{aligned}
&\frac{1}{T}\Big(\frac{1}{2}-\frac{3|V|L^2\gamma^2}{(1-\rho)\Big(1-\frac{6|V|L^2\gamma^2}{1-\rho}\Big)}\Big)\sum_{t=0}^{T-1}\mathbb{E}\Big\|\nabla f\Big(\frac{(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\Big)\Big\|^2\\
&\leq \frac{\sum_{t=0}^{T-1}\mathbb{E}\Big[f\Big(\frac{(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\Big)-f\Big(\frac{X_{t+1}\mathbf{1}_{|V|}}{|V|}\Big)\Big]}{\gamma T}+\frac{\gamma L\sigma^2}{2|V|}+\frac{L^2\gamma^2|V|(\sigma^2+3\varsigma^2)}{(1-\rho)\Big(1-\frac{6|V|L^2\gamma^2}{1-\rho}\Big)}+\frac{L^2\varphi d|V|\sum_{t=0}^{T-1}\phi_t^2}{2T(1-\rho)\Big(1-\frac{6|V|L^2\gamma^2}{1-\rho}\Big)}
\end{aligned}\tag{36}$$

Let

$$P_1:=\Big(\frac{1}{2}-\frac{3|V|L^2\gamma^2}{(1-\rho)P_2}\Big), P_2:=\Big(1-\frac{6|V|L^2\gamma^2}{1-\rho}\Big)$$

We can rewrite (36) as follows:

$$\begin{aligned}
&\frac{\sum_{t=0}^{T-1}\mathbb{E}\Big\|\nabla f\Big(\frac{(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\Big)\Big\|^2}{T}\\
&\leq \frac{\sum_{t=0}^{T-1}\mathbb{E}\Big[f\Big(\frac{(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\Big)-f\Big(\frac{X_{t+1}\mathbf{1}_{|V|}}{|V|}\Big)\Big]}{\gamma T P_1}+\frac{\gamma L\sigma^2}{2|V|P_1}+\frac{L^2\gamma^2|V|(\sigma^2+3\varsigma^2)}{(1-\rho)P_1P_2}+\frac{L^2\varphi d|V|\sum_{t=0}^{T-1}\phi_t^2}{2T(1-\rho)P_1P_2},
\end{aligned}\tag{37}$$

To further analyze the result of convergence of (37), we let $\gamma=\sqrt{\frac{1-\rho}{\mathcal{C}|V|L^2}}$ where $\mathcal{C}>0$ is a constant such that $P_1, P_2>0$. we obtain:

$$\frac{\sum_{t=0}^{T-1}\mathbb{E}\big\|\nabla f\big(\frac{(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\big)\big\|^2}{T}\leq\mathcal{O}\Big(\frac{\sum_{t=0}^{T-1}\mathbb{E}\Big[f\Big(\frac{(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\Big)-f\Big(\frac{X_{t+1}\mathbf{1}_{|V|}}{|V|}\Big)\Big]}{T(1-\rho)}\Big)+\mathcal{O}\Big(\frac{\varphi\mathcal{U}}{T(1-\rho)}\Big),\tag{38}$$

where $\mathcal{U}=\sum_{t=0}^{T-1}\phi_t^2$.

If no device perturbs local parameters with noises, then the spectral gap is different, say $(1-\rho')$. Assume the optimal solution is obtained at time T we can obtain the bound below:

$$\frac{\sum_{t=0}^{T-1}\mathbb{E}\big\|\nabla f\big(\frac{X_t\mathbf{1}_{|V|}}{|V|}\big)\big\|^2}{T}\leq\mathcal{O}\Big(\frac{\mathbb{E}\Big[f\Big(\frac{X_0\mathbf{1}_{|V|}}{|V|}\Big)-f(x^*)\Big]}{T(1-\rho')}\Big)\tag{39}$$

The theorem follows.

C. Proof to Corollary 1

Following the results in Theorem 1, we obtain the following inequality.

$$\begin{aligned}
&\frac{\sum_{t=0}^{T-1}\mathbb{E}\big\|\nabla f\big(\frac{(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\big)\big\|^2}{T}\\
&\overset{(a)}{\leq}\frac{\mathbb{E}\big[\sum_{t=1}^{T-1}L/2\big\|\frac{W_t\mathbf{1}_{|V|}}{|V|}\big\|^2+f\big(\frac{(X_0+W_0)\mathbf{1}_{|V|}}{|V|}\big)\big]}{T(1-\rho)}+\frac{\gamma L\sigma^2}{2|V|P_1}+\frac{L^2\gamma^2|V|(\sigma^2+3\varsigma^2)}{(1-\rho)P_1P_2}+\frac{L^2\varphi d|V|\mathcal{U}}{2T(1-\rho)P_1P_2},
\end{aligned}\tag{40}$$

where (a) follows from Assumption 2.

Also, the bound of the convergence rate without noises is as follows.

$$\begin{aligned}
&\frac{\sum_{t=0}^{T-1}\mathbb{E}\big\|\nabla f\big(\frac{X_t\mathbf{1}_{|V|}}{|V|}\big)\big\|^2}{T}\\
&\leq\frac{L\|x^*\|^2}{2T(1-\rho')}+\frac{\gamma L\sigma^2}{2|V|P_1}+\frac{L^2\gamma^2|V|(\sigma^2+3\varsigma^2)}{(1-\rho')P_1P_2}
\end{aligned}\tag{41}$$

Suppose that $\alpha=\frac{1-\rho}{1-\rho'}\geq 1$. Subtracting (40) with (41), we yield

$$
\begin{aligned}
&\frac{\sum_{t=0}^{T-1}\mathbb{E}\big\|\nabla f\big(\frac{(X_t+W_t)\mathbf{1}_{|V|}}{|V|}\big)\big\|^2}{T}-\frac{\sum_{t=0}^{T-1}\mathbb{E}\big\|\nabla f\big(\frac{X_t\mathbf{1}_{|V|}}{|V|}\big)\big\|^2}{T}\\
&\leq\frac{\sum_{t=1}^{T-1}L/2\big\|\frac{W_t\mathbf{1}_{|V|}}{|V|}\big\|^2+f\big(\frac{(X_0+W_0)\mathbf{1}_{|V|}}{|V|}\big)}{T(1-\rho)}+\frac{L^2\varphi d|V|\mathcal{U}}{2T(1-\rho)P_1P_2}-\frac{L\alpha\|x^*\|^2}{2T(1-\rho)}+\frac{(1-\alpha)[L^2\gamma^2|V|(\sigma^2+3\varsigma^2)]}{(1-\rho)P_1P_2}\\
&\overset{(a)}{\leq}\frac{\sum_{t=1}^{T-1}L/2\big\|\frac{W_t\mathbf{1}_{|V|}}{|V|}\big\|^2+f\big(\frac{(X_0+W_0)\mathbf{1}_{|V|}}{|V|}\big)}{T(1-\rho)}+\frac{L^2\varphi d|V|\mathcal{U}}{2T(1-\rho)P_1P_2}-\frac{L\alpha\|x^*\|^2}{2T(1-\rho)}\\
&=\mathcal{O}\bigg(\frac{\sum_{t=1}^{T-1}\big\|\frac{W_t\mathbf{1}_{|V|}}{|V|}\big\|^2+f\big(\frac{(X_0+W_0)\mathbf{1}_{|V|}}{|V|}\big)+\varphi\mathcal{U}-\alpha\|x^*\|^2}{T(1-\rho)}\bigg),
\end{aligned}
\tag{42}
$$

where (a) holds since $\alpha\geq 1$. The proof is completed.

D. Implementation Details regarding Regression Function and Neural Network

1) Regression Function: Sklearn in python is used to implement regression function. The function is trained to fits the training curves in our real dataset with hitting time $\mathcal{H}(G_c)$, and ratio of DP devices φ. Each data point in the real dataset is formulated as a three-tuple vector. For example, a vector $(100, 0.1, 300)$ indicates a communication topology with $\mathcal{H}(G_c)=100, \varphi=0.1$, requires 300 rounds achieve 70% accuracy on the benchmark CIFAR10. Mean square error (MSE) works as loss function to carry out gradient descent for updating the regression model. The learning rate is set to 0.001, and initial weights are randomly select a real number between 1 and 10.

2) Neural Network: Both two CLs have 64 5×5 channels and each layer is followed by a 3×3 max pooling with a stride of two and normalization. The first two FLs have 384 and 192 units (each of them with ReLu activation followed by one dropout), and the last FL is the final softmax output layer with 10 units. The initial model parameters follow the suggestions in [7], [8]. The learning rate, learning rate decay, number of local epochs, and local minibatch size, are set to 0.2, 0.99, 5, and 64, respectively.

E. Pseudocode of AutoTag

Algorithm 2 AutoTag

Input: The social network $G_s = (V, E_s)$, physical network $G_p = (V, E_p)$, communication time d_{ij} of each link $(i, j) \in E_p$, and computation time r_i of each device $i \in V$

Output: The communication topology $G_c = (V, E_c)$

Connectivity Guarantee Phase (CGP)

Collection of candidate solutions $\mathbb{S}$ is empty initially;
for all $0 \leq n \leq |V|$ **do**
  Candidate solution $E_c^n \leftarrow \emptyset$;
  $L \leftarrow V \times V$;
  The set of devices adopting DP $N \leftarrow \emptyset$;
  while $|E_c^n| < |V|$ **do**
    Link set $\mathcal{L} \leftarrow \arg\min_{e=(i,j)\in L}(deg_c(i) + deg_c(j))$;
    Link $l \leftarrow \arg\min_{e=(i,j)\in \mathcal{L}\cap E_s}(d_{ij} + \max\{r_i, r_j\})$;
    if link l does not exist **then**
      Link $l \leftarrow \arg\min_{e=(i,j)\in \mathcal{L}}(d_{ij} + \max\{r_i, r_j\})$;
    $L \leftarrow L \setminus \{l\}$;
    if $l \in E_s$ or $|N \cup V(l)| \leq n$ **then**
      $E_c^n \leftarrow E_c^n \cup \{l\}$;
      if $l \notin E_s$ **then**
        $N \leftarrow N \cup V(l)$;
  $G_c^n \leftarrow (V, E_c^n)$;
  Collection of candidate solutions $\mathbb{S} \leftarrow \mathbb{S} \cup \{G_c^n\}$;

Loner Connection Phase (LCP)

for all $0 \leq n \leq |V|$ **do**
  $L \leftarrow V \times V \setminus E_c^n$;
  while $|N| < n$ and $|L| > 0$ **do**
    Update m_{ij} for each pair $i, j \in V$ via eq. (11);
    Link $l \leftarrow \arg\max_{e=(i,j)\in L} SLS(i, j)$;
    $L \leftarrow L \setminus \{l\}$;
    if $l \in E_s$ or $|N \cup V(l)| \leq n$ **then**
      $E_c^n \leftarrow E_c^n \cup \{l\}$;
      if $l \notin E_s$ **then**
        $N \leftarrow N \cup V(l)$;
      $G_c^n \leftarrow (V, E_c^n)$;
      Collection of candidate solutions $\mathbb{S} \leftarrow \mathbb{S} \cup \{G_c^n\}$;

Network Expanding Phase (NEP)

for all $0 \leq n \leq |V|$ **do**
  $L \leftarrow V \times V \setminus E_c^n$;
  while $|L| > 0$ **do**
    Update m_{ij} for each pair $i, j \in V$ via eq. (11);
    Link $l \leftarrow \arg\max_{e=(i,j)\in L} CLS(i, j)$;
    $L \leftarrow L \setminus \{l\}$;
    if $l \in E_s$ or $|N \cup V(l)| \leq n$ **then**
      $E_c^n \leftarrow E_c^n \cup \{l\}$;
      if $l \notin E_s$ **then**
        $N \leftarrow N \cup V(l)$;
      $G_c^n \leftarrow (V, E_c^n)$;
      Collection of candidate solutions $\mathbb{S} \leftarrow \mathbb{S} \cup \{G_c^n\}$;

Snapshot Selection Phase (SSP)

$G_c \leftarrow \arg\min_{\mathcal{S}\in\mathbb{S}}(\mathcal{G}'(\mathcal{S}) \cdot \mathcal{L}(\mathcal{S}))$;
return G_c;

REFERENCES

[1] N. Agarwal *et al.*, "cpSGD: Communication-efficient and differentially-private distributed SGD," in *NeurIPS*, 2018.

[2] C. Dwork, A. Roth, *et al.*, "The algorithmic foundations of differential privacy.," *Foundations and Trends in Theoretical Computer Science*, vol. 9, no. 3-4, pp. 211–407, 2014.

[3] Y. Xiong, J. Xu, K. You, J. Liu, and L. Wu, "Privacy preserving distributed online optimization over unbalanced digraphs via subgradient rescaling," *IEEE Trans. on Control of Net. Sys.*, 2020.

[4] X. Zhang *et al.*, "Private and communication-efficient edge learning: a sparse differential gaussian-masking distributed SGD approach," in *ACM Mobihoc*, 2020.

[5] B. Jayaraman and L. Wang, "Distributed learning without distress: Privacy-preserving empirical risk minimization," *Advances in Neural Information Processing Systems*, 2018.

[6] X. Lian *et al.*, "Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent," in *NeurIPS*, 2017.

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

[8] J. Y. Yam and T. W. Chow, "A weight initialization method for improving training speed in feedforward neural network," *Neurocomputing*, vol. 30, no. 1-4, pp. 219–232, 2000.