Efficient Online Decentralized Learning Framework for Social Internet of Things

Cheng-Wei Ching, Hung-Sheng Huang, Chun-An Yang, Jian-Jhih Kuo, and Ren-Hung Hwang
Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi, Taiwan
Email: {g08410092, g08410048, u06115002}@ccu.edu.tw, {lajacky, rhhwang}@cs.ccu.edu.tw

***Abstract*—Online Decentralized Learning (ODL) is suitable for Internet-of-Things (IoT) devices since only parameter updates are exchanged with neighbors to avoid uploading private data to a central server and the training data is allowed to arrive at the devices sequentially. However, the current ODL frameworks cannot support the emerging Social IoT (SIoT) paradigm favorably since the SIoT devices exchange parameter updates with only trustworthy neighbors based on specific social relations (e.g., parental object relation and ownership object relation). Conversely, sharing parameter updates with untrustworthy neighbors could speed up the training process but may violate social relations. Differential privacy (DP) is thus used 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. Thus, in this paper, we innovate an ODL framework ODLF-PDP to allow only a part of devices to engage DP (i.e., partially DP) to improve training performance. Then, an algorithm BeTTa is proposed to build an adequate communication topology based on the interplay among the social networks, physical networks, and DP. Last, the experiment results manifest that ODLF-PDP saves more than 20% physical training time compared to the current frameworks via the benchmark of MNIST.**

I. Introduction

Recently, Social Internet of Things (SIoT) with *Artificial Intelligence (AI) on chips* is a promising network paradigm. SIoT devices monitor the environment, collect the data, and interact and establish relationship with each other [1]. They 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 data from SIoT devices to a single central server for training is impracticable. Fortunately, online decentralized learning (ODL) can overcome the privacy issue, where each device executes on-device training but exchanges parameter updates with its *neighbors* based on a given *communication topology* [3]–[6].[1] That is, each device acts as both a training unit and a parameter aggregator at the same time, and the central server fades away.

ODL also allows the data for training to become available *in sequential order*. This property is suitable for user devices' training since it enables devices with limited storage to train models and collect data simultaneously [3]. However, the state-of-the-art approaches of ODL usually make communication topologies a ring, torus, or an expander directly [4], [7], [8] and ignore two indispensable factors in SIoT-based environment. One is *socially topological restriction* since two SIoT devices are not allowed to communicate if they have no social tie. The other is *communication bottleneck among devices*. The SIoT devices are typically available to distinct communication manners, such as Wi-Fi, LoRa, and so forth [1]. An arbitrary constructed communication topology may lead to time-consuming communication bottleneck among the SIoT devices and prolong the training time. Besides, current ODL usually assumes that the batch size of each training device is the same (e.g., 16, 32) and ignores the benefits of better convergence rate derived from adaptive (sufficiently large) batch sizes [3].

Therefore, to jointly deal with the above issues, in this paper, we propose the **O**nline **D**ecentralized **L**earning **F**ramework with **P**artially **D**ifferential **P**rivacy (ODLF-PDP), an innovative ODL training framework for SIoT-based environments. To take an overview of devices' *social network*, *SIoT platforms* are employed to arrange the communication topologies since SIoT devices and their social profiles are registered in SIoT platforms (e.g., iSapiens) [9]. The SIoT platforms feature the *assessment and management of social relationship and trustworthiness* among devices such that the relations can be further established *autonomously* based on the interactions and social conditions [9]. Therefore, ODLF-PDP consists of 1) a set of SIoT devices with *distinct computing units*, and *communication capability* and 2) an SIoT platform. The SIoT devices take part in a given training task (e.g., class classification), and the SIoT platform arranges the communication topology of SIoT devices.

To overcome the socially topological restriction, ODLF-PDP exploits an additional link between two SIoT devices without social relation and secure such a link by differential privacy (DP). That is, ODLF-PDP enables parameter exchanges between SIoT devices without social relation but does not compromise privacy when coordinating SIoT devices to train models collaboratively. To fully utilize ODLF-PDP, the **B**atch-Size-Adaptive Time-**e**fficient **T**opology Cons**t**ruction **A**lgorithm (BeTTa) is proposed to make the SIoT platform construct a fit and effective communication topology for ODL (detailed in Section III-A). BeTTa selects certain SIoT devices to traverse more data samples based on their *computing power of devices* and *transmission conditions* of the communication topology while not prolonging the overall training time.

The contributions of this paper are summarized as follows. On the theoretical side, we rigorously prove that 1) ODLF-PDP retains the same order of regret bound in [3], [10], [11] even in the more complicated scenarios and that 2) higher traverses of data samples improve the regret bound of ODLF-PDP. On the experimental side, we compare ODLF-PDP with two state-of-the-art ODL methods, PDOO [10] and DABMD [3]. The extensive experiment results show ODLF-PDP save the physical training time by at least 20%.

II. Preliminaries and Motivation

A. Mini-batch Online Decentralized Learning (MBODL) [3]

Given a connected communication topology $G_c = (V, E_c)$, each device $i \in V$ makes a decision parameter x_i^t in the

[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 ODL process.

constrained set $\mathcal{X} \in \mathbb{R}^m$ at round $t \in T$. Compared with offline decentralized optimization, the data samples arrive over time. Let $\xi_{i,t}^l$ denote the l-th data sample at round t executed locally by device i, where $l = 1, ..., \mathcal{B}_i^t$, and $\mathcal{B}_i^t$ is so-called mini-batch size of device i at round t. All the devices are given a local cost function $f_i^t : \mathcal{X} \to \mathbb{R}$, i.e.,

$$f_i^t(x) = \sum_{l=1}^{\mathcal{B}_i^t} f(x, \xi_{i,t}^l). \tag{1}$$

The mini-batch size is usually identical over each device in the literature [4], [8], [10], [12]. MBODL focuses on the global cost function, denoted by $f_t : \mathcal{X} \to \mathbb{R}$ at round t and the global cost function is defined based on the local cost functions, i.e.,

$$f_t(x) = \sum_{i \in V} f_i^t(x). \tag{2}$$

Then, the local decision parameter of each device at time t is updated as follows

$$x_i^{t+1} = \mathcal{P}_{\mathcal{X}}\Big(\sum_{j \in V} a_{ij} x_j^t - \eta_t \mathrm{g}_i^t \Big), \tag{3}$$

where $\eta_t \in (0, 1]$ is step size at time t, a_{ij}, an entry of communication matrix $\mathcal{A}(G_c)$ (see Definition 1) derived from communication topology G_c, g_i^t, gradient updates with respect to x_i^t, and $\mathcal{P}_{\mathcal{X}}(\cdot)$, the projection operator of a vector onto $\mathcal{X}$.

Definition 1 (Lazy-Metropolis-based Communication Matrix [8])**.** 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 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.

Also, the goal of MBODL is to minimize accumulated cost over total rounds T. To measure the quality of local decision parameters, we introduce the notion of regret [10], [11].

Definition 2. Suppose there exists a fixed optimal solution x^* to (2) of all time T. The regret of device $j \in V$ is defined by

$$\mathbb{E}[Reg_j^T] = \sum_{t \in T} \mathbb{E}[f_t(x_j^t)] - \sum_{t \in T} f_t(x^*). \tag{5}$$

A desired algorithm for online decentralized optimization shrinks the gap between the decision parameters (i.e., smaller regret) determined in online fashion and the offline counterpart.

B. Differential Privacy (DP)

Definition 3 (Differential Privacy [13])**.** A randomized algorithm $\mathbb{A} : \mathcal{D} \to \mathcal{R}$ with domain $\mathcal{D}$ and range $\mathcal{R}$ is said to be ϵ-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

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

where $\epsilon > 0$ is the privacy budget.

Simply put, ϵ should be kept low if the privacy level is highly demanded. However, higher privacy level sacrifices the accuracy of optimization problem (2). Therefore, we need a factor, sensitivity (see Definition 4), to determine how much noise should be generated to perturb the process of optimization and guarantee the privacy level at the same time.

Definition 4 (Sensitivity [10], [11])**.** The sensitivity of a randomized algorithm $\mathbb{A}$ at iteration $t \geq 0$ is defined as follows

$$\Delta_t = \sup_{d, d'} \|\mathbb{A}_t(d) - \mathbb{A}_t(d')\|_1. \tag{7}$$

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'.

C. Topological and Mini-batch-size Impact on MBODL

Recall that the local decision parameters are influenced by the neighbors. Many works have shown that the convergence and regret are highly influenced by the connectivity of communication topologies [3], [4], [7], [8]. To define the connectivity, we review the concept of spectral gap.

Definition 5 (Spectral Gap)**.** Let $\lambda_i(\mathcal{A}(G_c))$ denote the eigenvalue with the i^{th} largest absolute value of matrix $\mathcal{A}(G_c)$. Following Definition 1, $\mathcal{A}(G_c)$ has singular values ranging $1 = |\lambda_1(\mathcal{A}(G_c))| > |\lambda_2(\mathcal{A}(G_c))| \geq \cdots \geq |\lambda_{|V|}(\mathcal{A}(G_c))|$ and spectral gap $\delta(G_c) = 1 - |\lambda_2(\mathcal{A}(G_c))| \in (0, 1]$.

The spectral gap can be a metric to measure the connectivity of a topology. Higher spectral gap represents better connectivity. However, the communication topologies are highly limited due to the social network of SIoT. Inspired by the mechanism of DP used in [10], [11], where all devices perturb local parameters before exchanging with their neighbors in order to guarantee privacy. We propose the mechanism of *partially differential privacy*, where partial devices adopt DP so that the links between two DP devices can be exploited in communication topologies. The method can increase the spectral gap of communication topologies and two SIoT devices that have no relation (i.e., no trust link between them in the social network) can exchange parameters without worrying about possible model inversion attacks [14].

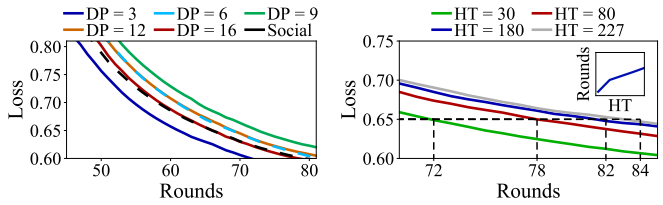
In the literature, the mini-batch size is typically fixed over time while sufficiently large mini-batch size can usually improve the performance [15], [16]. In this sense, we set a basic mini-batch size[2] for each SIoT device and then increase some SIoT devices' mini-batch size adaptively without exacerbating the communication bottleneck. In the following, we first prove the relation between spectral gap in ODL.

Theorem 1 (Convergence of local decision parameters)**.** Suppose that the three assumptions hold. We have the inequality

$$\mathbb{E}[\|\bar{x}^t - x_i^t\|] \leq \mathcal{O}\Big(\frac{\varphi}{\epsilon \delta(G_c)}\Big), \tag{8}$$

where $\varphi \in [0, 1]$ presents the ratio of the number of DP devices to that of all devices and $\bar{x}^t = \frac{1}{|V|} \sum_{i \in V} x_i^t$.

[2]Many works [15], [16] implement model training with a basic mini-batch size (e.g., 32 or 64) depending on the target tasks (e.g., class classification).



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

Fig. 1. (a) Effect of number of DP devices ranging $[0, 16]$ on loss in a 16-node network with $\epsilon = 1$. (b) Effect of hitting time ranging $[30, 227]$ on loss in a 16-node network.

TABLE I
TRAINING TIME FOR DIFFERENT MINI-BATCH SIZES (SEC)

Target Accuracy	65%	68%	70%	72%
mini-batch size $\mathcal{B}$ = 256	742.2	839.8	904.4	1031.7
mini-batch size $\mathcal{B}$ = 64	753.1	861.5	1003.8	1254.4

Theorem 1 is similar to the result of [10], but the latter result is not based on partially differential privacy and thus cannot be applied to ODLF-PDP. Also, we bound the regret.

Theorem 2. Let T denote the number of executed iterates for ODL. The expected regret satisfies:

$$\mathbb{E}[REG_j^T] \leq \mathcal{O}\left(\sqrt{T}\left(\frac{\varphi^2}{\epsilon^2 \delta(G_c)^2 \mathcal{B}_{avg}}\right)\right), \tag{9}$$

where $\varphi \in [0, 1]$ denotes the ratio of the number of DP devices to that of all devices, ϵ is the privacy budget, $\delta(G_c)$ is the spectral gap, and $\mathcal{B}_{avg}$ is the average mini-batch size across all devices and rounds.

The proofs of Theorems 1 and 2 are presented in Appendix A. Note that our regret bound achieves the same order of that in [3], [10], [11] even if we consider more complicated scenarios. With Theorems 1 and 2, we can obviously see that a higher average mini-batch size and a higher spectral gap lead to better convergence and regret. A higher φ may degrade the performance but can increase the spectral gap. To see how two factors affect each other, we conduct motivating experiments to emphasize three control factors. We implemented the ODL framework [3] under communication topologies based on a fixed mini-batch size with different numbers of DP devices but derived from the same social network with 16 devices extracted from Santander (a city in Spain) [2].[3] The results are depicted in Fig. 1(a). It's obvious to see that a suitable number of devices that adopts DP can obtain faster convergence rate, which matches the results in Theorem 2. The following proposition shows that the mechanism of partially DP guarantee ϵ-differential privacy for the DP devices:

Proposition 1. Suppose that the above three assumptions hold. A randomized algorithm guarantees ϵ-differentially privacy for the selected DP devices in each round if the selected DP devices perturb its parameters by adding Laplace noises with variance $2(\sigma_t)^2$, where $\sigma_t = \Delta_t/\epsilon, \forall t \in T$, and $\epsilon > 0$.

The proof of Proposition 1 can be easily derived from [10]. Next, we follow the same implementation settings but train the model with the same communication topology. The effects of the different mini-batch sizes on the training performance are summarized in Table I. The result with a larger mini-batch size can achieve better convergence rate at the expense of training time, which means that its regret is lower than that with a smaller mini-batch size (i.e., Theorem 2).

[3]Notice that the social network is built based on *ownership object relation* in the Santander dataset

TABLE II
TRAINING TIME IN DIFFERENT PHYSICAL NETWORKS

Target Accuracy	65%	70%	75%	80%
Number of Rounds	27	33	49	92
Computation Time (sec)	567.6	693.7	1030.1	1934.1
Communication Time in Net1 (sec)	4.1	4.9	7.4	21.2
Communication Time in Net2 (sec)	256.5	313.5	465.5	874.1

Apart from the factors that can be mathematically proved, the hidden impact of physical networks also holds sways of the training time of ODL. In the next experiment with the same settings, we constructed two different physical networks, where the mini-batch size is fixed and each device is equipped with equal computing power. The effects of the two different physical networks on communication time are shown in Table II, where the two physical networks Net1 and Net2 with the same communication topology have different communication bottleneck of 0.15 and 9.5 seconds based on different transmission manners (e.g., WiFi, LoRa) [17], [18]. The results show the training time in different physical networks can be largely influenced (i.e., $313.5 - 4.9 = 308.6$ seconds) by the communication bottleneck among the devices.

III. METHODOLOGY

The framework is shown in Fig. 2. The designs of BeTTa and ODLF-PDP are detailed in Sections III-A and III-B.

A. The Design of BeTTa

To construct an efficient communication topology G_c with a high spectral gap, we should balance *global iterate* $\mathcal{G}(G_c)$ (i.e., the number of rounds to achieve a specific accuracy) and *local iterate* $\mathcal{L}(G_c)$ (i.e., the time required for each round) for exchanging local model updates among devices to reduce physical training time. Therefore, BeTTa aims at minimizing the physical training time $\mathcal{G}(G_c) \cdot \mathcal{L}(G_c)$. Also, the additional mini-batch sizes for each SIoT device should be determined as well. Since the link in G_c requiring the most time in G_p will be the physical bottleneck to prolong overall training procedure, we can define local iterate as follows.

Definition 6 (Local Iterate)**.** The local iterate is defined as the maximum communication and computation time of the links E_c in communication topology G_c, i.e.,

$$\mathcal{L}(G_c) = \max_{(i,j) \in E_c} (d_{ij} + \max\{r_i, r_j\}), \tag{10}$$

where d_{ij} is the communication time between two SIoT devices $i, j \in V$ and r_i is the computation time for a *basic mini-batch size* of device $i \in V$.

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. 3(a). Take the topology in Fig. 3(b) for example. Suppose the basic mini-batch size is set to 32. By eq. (10), link $\overline{DE}$ dominates the local iterate, which is $45 + 7 = 52$. ■

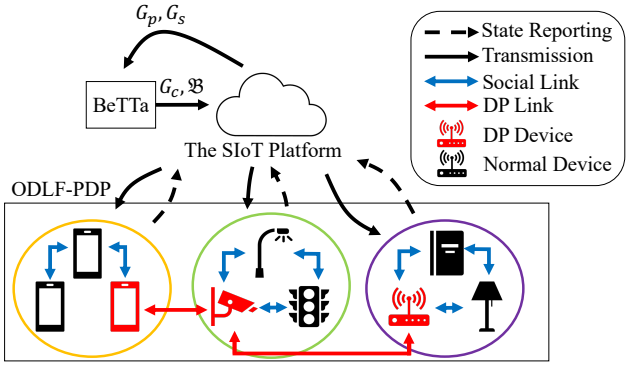


Fig. 2. Architecture of ODLF-PDP and BeTTa.

However, explicitly defining global iterate $\mathcal{G}(G_c)$ is non-trivial [19] so we make use of the notion of the *hitting time* $\mathcal{H}(G_c)$ instead since the hitting time of a communication topology can be calculated in polynomial time and provides a useful upper bound of global iterate as follows:

Proposition 2. In [8], Proposition 4 shows the relation between global iterate and spectral gap is $\mathcal{G}(G_c) \propto \frac{1}{\delta(G_c)}$, and Proposition 5 further bounds $\frac{1}{\delta(G_c)}$ by $\mathcal{O}(\mathcal{H}(G_c))$, where $\mathcal{H}(G_c)$ is the hitting time of G_c with the matrix established by (4). Therefore, we yield the following induction

$$\mathcal{G}(G_c) \propto \frac{1}{\delta(G_c)} = \mathcal{O}(\mathcal{H}(G_c)). \tag{11}$$

Following Proposition 2, we need to consider the hitting time $\mathcal{H}(G_c)$ (see Definition 7) to predict the global iterate.

Definition 7 (Hitting Time)**.** Given a communication matrix $\mathcal{A}(G_c)$ calculated by (4), the entries of relevant hitting time matrix $\mathcal{M}(G_c^t) \in \mathbb{R}^{|V|\times|V|}$ are defined as

$$m_{ij} = \begin{cases} 0, & \text{if } i = j, \\ 1 + \sum_{k\in V, k\neq j} a_{ik} \cdot m_{kj}, & \text{otherwise}, \end{cases} \tag{12}$$

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., $\mathcal{H}(G_c) = \max_{i,j\in V} m_{ij}$.

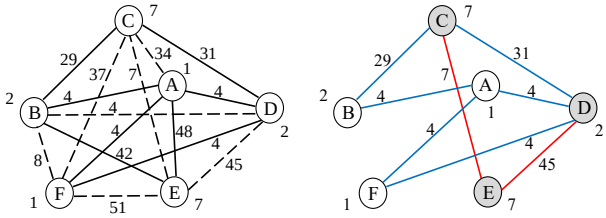
Remark that the hitting time between i and j is *bidirectional* and the rationale behind $\mathcal{H}(G_c)$ is detailed in Example 2.

Example 2. This example shows the calculation of hitting time with the network in Fig. 3(a), where matrix $\mathcal{M}$ is 6×6. Take the topology in Fig. 3(b) for example. By eq. (12), $m_{AA} = 0$, $m_{BA} = 1 + \frac{2}{3}m_{BA} + \frac{1}{6}m_{CA}$, $m_{CA} = 1 + \frac{1}{6}m_{BA} + \frac{13}{24}m_{CA} + \frac{1}{8}m_{DA} + \frac{1}{6}m_{EA}$, and the followings are omitted. Thus, there are 36 variables attained from 36 equations, and the hitting time is 26, where entire $\mathcal{M}$ is shown Appendix B. ■

By Theorem 2 and Proposition 2, the expected regret of ODLF-PDP is proportional to $\left(\frac{\varphi}{\epsilon\delta(G_c)}\right)^2$, where $\frac{1}{\delta(G_c)}$ is bounded by $O(\mathcal{H}(G_c))$. Therefore, BeTTa constructs the communication topology G_c with the following alternative and asymptotic goal — minimize the *pseudo training time*:

$$\text{minimize} \quad \mathcal{H}(G_c) \cdot \mathcal{L}(G_c) \cdot \left[1 + \left(\frac{\varphi}{\epsilon\delta(G_c)}\right)^2\right]. \tag{13}$$

BeTTa constructs a candidate solution for each possible number of DP devices and then picks the best one among them



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

Fig. 3. (a) Solid and dashed lines represent trust and untrust links in the social network G_s, respectively. The number next to each link and node denotes the communication time between two devices and computation time of device for a basic mini-batch size with respect to a target task in physical network G_p, respectively. (b) shows an example of communication topology G_c. The blue and red links represent the trust and untrust links selected in G_c, respectively, and the gray nodes denote the DP devices.

by virtue of eq. (13). Thus, there are at most $\lfloor|V|\rfloor + 1$ candidate solutions, $G_c^0, ..., G_c^{\lfloor|V|\rfloor}$. Each candidate solution runs the following five steps.

1) Connectivity Assurance Step (CAS): All the involved devices must be connected in G_c^n, where $0 \leq n \leq \lfloor|V|\rfloor$. Also, a more regular graph usually has a smaller hitting time. Thus, CAS 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 untrust link (i.e., not in G_s) if the number of DP devices is no greater than n. Otherwise, CAS will discard it. Note that G_c^n is connected after CAS since G_s is connected.

2) Solitary Connection Step (SCS): The solitary 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, SCS 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, SCS iteratively selects the pair of devices with the maximum *social solitary score* (SSS) defined as follows, where $deg_s(i)$ is the degree of device $i \in V$ in G_s.

$$SSS(i, j) = \frac{\max\{m_{ij}, m_{ji}\}}{(d_{ij} + \max\{r_i, r_j\})\ deg_s(i)\ deg_s(j)}. \tag{14}$$

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 Augmenting Step (NAS): To address the trade-off between global and local iterates, NAS 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, NAS iteratively selects the pair with the maximum *network solitary score* (NSS) defined as follows, where $deg_c(i)$ denotes the degree of device $i \in V$ in G_c^n.

$$NSS(i, j) = \frac{\max\{m_{ij}, m_{ji}\}}{(d_{ij} + \max\{r_i, r_j\})\ deg_c(i)\ deg_c(j)}. \tag{15}$$

Subsequently, BeTTa picks the best possible candidate solution with the following phase.

4) Candidate Selection Step (CSS): For each n, CSS selects the one with the minimum pseudo training time by (13) 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 $G_c^0, ..., G_c^{\lfloor |V| \rfloor}$ to be the solution G_c.

Lastly, BeTTa outputs the communication topology selected in SSP and determines the additional mini-batch sizes for each SIoT device according to the following phase:

5) Mini-batch-size Expanding Step (MEP): To increase the mini-batch sizes of devices without increasing the training time, we find two devices $\hat{i}, \hat{j}$ that dominate the local iterate of the constructed G_c, i.e.,

$$(\hat{i}, \hat{j}) = \underset{(i,j) \in E_c}{\arg\max} (d_{ij} + \max\{r_i, r_j\}). \tag{16}$$

Each device $k \in V$ other than $\hat{i}, \hat{j}$ can increase its mini-batch size $\mathcal{B}_k$ without prolonging the local iterate, i.e.,

$$\mathcal{B}_k = \frac{d_{\hat{i}\hat{j}} + \max\{r_{\hat{i}}, r_{\hat{j}}\} - \max_{l \in \mathcal{N}(k)} d_{k,l}}{b_k}, \tag{17}$$

where b_k is the computation time of device k for a single data sample and $\mathcal{N}(k)$ is the neighbors of device k.

B. The Design of ODLF-PDP

ODLF-PDP employs the same global cost functions defined by eq. (2) but the local cost function in the same form of eq. (1). Compared with the traditional methods, ODLF-PDP can support the volatile mini-batch sizes over devices. To this end, ODLF-PDP employs BeTTa to construct the communication topology and configure the mini-batch size of each device such that the training fits the conditions of the social and physical networks. It also considers the computing power of each device and makes sure that the additional mini-batch sizes of devices hardly increase the overall training time jointly.

In addition, since some devices adopt DP to perturb local parameters before exchanging with their neighbors, ODLF-PDP modifies the update rule of local decision parameters for DP devices, which is originally defined by (3), yielding

$$x_i^{t+1} = \mathcal{P}_{\mathcal{X}}\Big(\sum_{j \in V} a_{ij}(x_j^t + w_j^t) - \eta_t \mathrm{g}_i^t \Big), \tag{18}$$

where w_j^t is noise drawn from Laplace distribution that satisfies DP constraints at time t if device i is asked to adopt DP. The non-DP devices follows the original update rule of local decision parameters (3). Another insightful distinction is that a_{ij}, the entry of communication matrix $\mathcal{A}(G_c)$ derives from the subtly-designed communication topology G_c. The pseudocode of ODLF-PDP is presented Appendix C.

IV. Performance Evaluation

A. Evaluation Setup

We compare ODLF-PDP with PDOO [10], and DAMBD [3]. In particular, DAMBD and PDOO *directly* use social networks as communication topologies for training and exchanging parameters since none of them considers the construction of communication topologies. The difference between two methods is that PDOO requires *all the training devices* to perturb local decision parameters and DAMBD enables the training devices to adopt the dynamic mini-batch-size training. The settings of experiment is detailed as follows.

1) Dataset: We adopt the well-known dataset MNIST with 60,000 images. For the distribution of social relations and positions of devices, the *real-world* dataset Santander [2], 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) Environment: The *computation time* of the SIoT devices is estimated according to the GFLOPS benchmark of *Raspberry Pi Model B series*, RPi2, RPi3, and RPi4, which requires 77s, 32s, and 14s per local round of training, respectively. The distribution of computing power of devices follows normal distribution over three benchmarks. The *communication time* for transmitting 42-KB 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) [17]. If not, the devices communicate over LoRa whose data rate is at most 37.5 Kbps [18] since Wi-Fi can only cover some 100m [17].

3) Implementation Details: 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 Gowalla randomly. We extract 50,000 images from MNIST and distribute them to devices evenly but each device only has 300 data samples at first and $10 \sim 30$ data samples arrive at the devices each round. The rest of images are for testing. The cost function is defined as follows according to [3]:

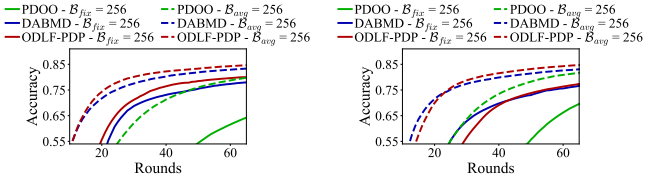
$$f(x, (s_i, y_i)) = \log(1 + \exp(-y_i x^T s_i)), \tag{19}$$

where x is the local decision parameter and s_i and y_i denote a data sample and its label, respectively. The goal of the devices is to find x^* as soon as possible when training data arrives in a sequential order [3]. The basic learning rate and learning rate decay are set to 0.01, and 0.99, respectively, and the privacy budget $\epsilon = 1$ follows the settings in [13]. Each implementation result is averaged over 10 trials.

B. Performance on Convergence Rate

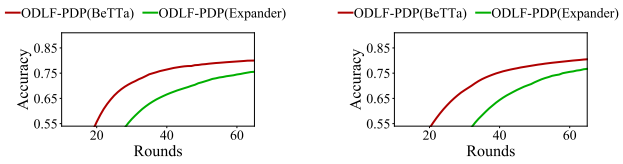
For a fair comparison, we first assume that the mini-batch sizes of all methods are identical. Fig. 4 shows the performance of three methods for different numbers of devices, and the same fixed and average mini-batch sizes (i.e., $\mathcal{B}_{fix}$ and $\mathcal{B}_{avg}$). Despite of the same setting, ODLF-PDP converges faster and achieves a higher accuracy than the other two since the communication topologies are subtly-constructed by making good use of DP. The performance of dynamic mini-batch sizes is better than that of the fixed one, which matches the experiment results in [3]. We also compare ODLF-PDP with a classic method for constructing communication topologies, expander graph [20], [21], usually adopted in ODL-like frameworks [3], [7]. With the same social and physical networks, BeTTa outperforms expander graph as shown in Fig. 5.

The effect of different average mini-batch sizes determined for the network status by different methods is shown in Fig. 6. DAMBD considers the computation time of each device to raise the average mini-batch size so the average mini-batch size is 64 and PDOO does not consider mini-batch sizes so we make it 64 as well. ODLF-PDP can obtain a higher average mini-batch size since most devices that do not dominate the communication and computation time can increase the average



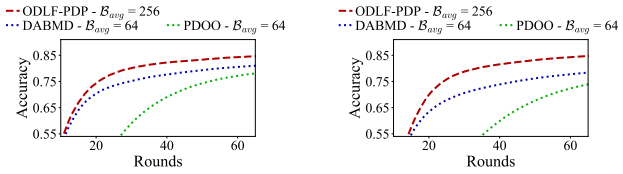
(a) Accuracy of $|V| = 16$ (b) Accuracy of $|V| = 32$

Fig. 4. Performance of three methods for different numbers of devices, the same fixed and average mini-batch sizes.



(a) Accuracy of $|V| = 16$ (b) Accuracy of $|V| = 32$

Fig. 5. Performance of ODLF-PDP(BeTTa) and ODLF-PDP(Expander) for different numbers of devices but the same mini-batch size.



(a) Accuracy of $|V| = 16$ (b) Accuracy of $|V| = 32$

Fig. 6. Effect of different numbers of devices. Note that the average mini-batch size $\mathcal{B}_{avg}$ is determined according to the network status by each method.

mini-batch sizes without compromising the overall training time. Therefore, ODLF-PDP converges faster and achieves a higher accuracy than DAMBD and PDOO. Please refer to Appendix D for the results regarding *loss*.

C. Performance on Physical Training Time

Table III shows the *physical training time* of three methods to a specific accuracy based on 16 and 32 devices. The methods follow the same social and physical networks to construct their communication topologies, where the socially topological restriction is considered. Take the 16-device case for example, BeTTa takes the similar amount of time to reach accuracy of 65% as does DAMBD but far less than does PDOO. The gap expands drastically for higher target accuracy (e.g., 2.7x and 1.6x). Note that the gap between ODLF-PDP and PDOO narrows down since the noises shrink due to the sensitivity (see Definition 4). ODLF-PDP requires much less physical training time than the other two in spite of doubling the devices. The results imply that ODLF-PDP is highly practical for ODL to accelerate the training procedure.

V. Conclusion

In this paper, we presented ODLF-PDP, an efficient training framework for ODL in SIoT-based scenarios. ODLF-PDP can overcome the slow training convergence due to the Communication bottleneck among devices because its component ODLF-PDP employs *partially* DP to relax the Communication bottleneck among devices and use some untrust links in social networks. Then, BeTTa is designed to construct an empirically efficient communication topology. Last, the extensive experiment results show that ODLF-PDP saves more than 20% of physical training time compared to the state of the art.

TABLE III
Physical Training Time with 16 and 32 devices (sec)

Target Accuracy	65%	70%	75%	80%
ODLF-PDP-16	822.3(1x)	953.9(1x)	1250.1(1x)	2138.1(1x)
DAMBD-16	872.1(1.1x)	1065.9(1.1x)	1582.7(1.3x)	2971.6(1.4x)
PDOO-16	2277.1(2.7x)	2752.2(2.8x)	3534.7(2.8x)	4893.7(2.3x)
ODLF-PDP-32	1469.3(1x)	1707.5(1x)	2223.7(1x)	3534.2(1x)
DAMBD-32	1364.1(0.9x)	1694.8(1x)	2356.3(1.1x)	4547.2(1.3x)
PDOO-32	2454.9(1.6x)	2746.1(1.6x)	3245.4(1.5x)	4493.7(1.3x)

References

[1] C.-H. Wang, J.-J. Kuo, D.-N. Yang, and W.-T. Chen, "Collaborative social internet of things in mobile edge networks," *IEEE Internet of Things J.*, vol. 7, no. 12, pp. 11 473–11 491, 2020.

[2] C. Marche, L. Atzori, V. Pilloni, and M. Nitti, "How to exploit the social Internet of Things: Query generation model and device profiles' dataset," *Comput. Netw.*, vol. 174, p. 107248, 2020.

[3] N. Eshraghi and B. Liang, "Distributed online optimization over a heterogeneous network," in *PMLR ICML*, 2020.

[4] A. Koloskova, S. Stich, and M. Jaggi, "Decentralized stochastic optimization and gossip algorithms with compressed communication," in *PMLR ICML*, 2019.

[5] C.-W. Ching, C.-K. Yang, Y.-C. Liu, C.-W. Hsu, J.-J. Kuo, H.-S. Huang, and J.-F. Lee, "Energy-efficient link selection for decentralized learning via smart devices with edge computing," in *IEEE GLOBECOM*, 2020.

[6] J.-J. Kuo, C.-W. Ching, H.-S. Huang, and Y.-C. Liu, "Energy-efficient topology construction via power allocation for decentralized learning via smart devices with edge computing," *IEEE Trans. on Green Comm. and Net.*, 2021.

[7] A. Nedić and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. Automat. Contr.*, vol. 54, no. 1, pp. 48–61, 2009.

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

[9] 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.

[10] Y. Xiong *et al.*, "Privacy preserving distributed online optimization over unbalanced digraphs via subgradient rescaling," *IEEE Trans. Control. Netw. Syst.*, vol. 7, no. 3, pp. 1366–1378, 2020.

[11] J. Zhu, C. Xu, J. Guan, and D. O. Wu, "Differentially private distributed online algorithms over time-varying directed networks," *IEEE Trans. on Signal and Info. Proc. over Net.*, vol. 4, no. 1, pp. 4–17, 2018.

[12] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *COMPSTAT*, 2010.

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

[14] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the GAN: information leakage from collaborative deep learning," in *ACM CCS*, 2017.

[15] N. S. Keskar *et al.*, "On large-batch training for deep learning: Generalization gap and sharp minima," *ICLR*, 2017.

[16] E. Hoffer, I. Hubara, and D. Soudry, "Train longer, generalize better: closing the generalization gap in large batch training of neural networks," in *NeurIPS*, 2017.

[17] S. R. Pokhrel, H. L. Vu, and A. L. Cricenti, "Adaptive admission control for iot applications in home WiFi networks," *IEEE Trans. on Mob. Comp.*, vol. 19, no. 12, pp. 2731–2742, 2019.

[18] R. Ghanaatian, O. Afisiadis, M. Cotting, and A. Burg, "Lora digital receiver analysis and implementation," in *IEEE ICASSP*, 2019.

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

[20] J. H. Kim and V. H. Vu, "Generating random regular graphs," in *ACM STOC*, 2003.

[21] 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," in *IEEE ICCCN*, 2021.

APPENDIX

A. Theoretical Analysis

To theoretically show the interplay among the spectral gap, mini-batch sizes, and the mechanism of partially differential privacy on the convergence and regret of MBODL, we require **three necessary assumptions** as follows. For ease of presentation, we let $\|\cdot\|$ represent l^2-norm if no specifically stated.

1) The gradients g_i^t of all the local cost functions f_i with respect to parameter $x \in \mathcal{X}$ are bounded, yielding
$$L_1 \geq \|\mathrm{g}_i^t\|, \qquad \forall i \in V, t \in T. \tag{20}$$

2) The objective functions of each device f_i are differentiable and L_3-strongly convex for $L_3 \geq 0$ if
$$\nabla f_i(x)(y-x) \leq f_i(y) - f_i(x) - \frac{L_2}{2}\|y-x\|^2. \tag{21}$$

3) We assume that the initial local parameters of each device $x_i^0 = 0, \forall i \in V$, which simplifies the analysis and proof.

The above assumptions are rational and typical to offer theoretical bound of ODL [3], [10]. The following proposition shows the convergence of varying doubly stochastic matrix.

Proposition 3. Let $\mathcal{A}^t$ denote $\mathcal{A}$ to the power of t. There exist constants $\gamma \geq 1$ and $\lambda \in (0,1)$ such that for any $i, j \in V$
$$|(\mathcal{A}^t)_{ij} - \frac{1}{|V|}| \leq \gamma\lambda^t, \tag{22}$$
where $(\mathcal{A}^t)_{ij}$ denotes the entry of $\mathcal{A}^t$ on the i^{th} row and j^{th} column of doubly stochastic matrix $\mathcal{A}^t$.

Proposition 3 is derived from [8] and the proof is detailed in [7]. Now, we are ready to show the bounds of the convergence rate and regret.

Proof of Theorem 1. We bound the sensitivity as follows first.

Lemma 1. Assume that (20) hold, we yield
$$\Delta_t \leq 2\eta_t\sqrt{m}L_1,$$
where m denotes the dimension of decision parameter x.

The proof of Lemma 1 can be found in [10]. Let $u_i^t = x_i^t - \sum_{j\in V} a_{ij}y_j^{t-1}$ where y_j^{t-1} is defined in Algorithm 1. First, we consider local parameter at time t.
$$x_i^t = u_i^t + \sum_{j\in V} a_{ij}x_j^{t-1} + \sum_{j\in V} a_{ij}w_j^{t-1} \tag{23}$$
Recursively applying to (23), we obtain
$$x_i^t = \sum_{j\in V}\mathcal{A}_{ij}^t x_j^0 + \sum_{s=1}^{t}\sum_{j\in V}\mathcal{A}_{ij}^{t-s}u_j^s + \sum_{s=1}^{t}\sum_{j\in V}\mathcal{A}_{ij}^{t-s+1}w_j^{s-1}, \tag{24}$$
where $\mathcal{A}^0 = \mathcal{I}$. Then, we consider the average parameter $\bar{x}^t$
$$\bar{x}^t \overset{(a)}{=} \frac{1}{|V|}\sum_{j\in V} x_j^0 + \frac{1}{|V|}\sum_{s=1}^{t}\sum_{j\in V} u_j^s + \frac{1}{|V|}\sum_{s=1}^{t}\sum_{j\in V} w_j^{s-1},$$
where (a) follows from the property of doubly stochastic matrix $\mathcal{A}$ and (24). Afterwards, we minus two terms and obtain
$$\begin{aligned}\|\bar{x}^t - x_i^t\| \leq& \sum_{j\in V}\Big\|\frac{1}{|V|} - \mathcal{A}_{ij}^t\Big\|\|x_j^0\| \\ &+ \sum_{s=1}^{t}\sum_{j\in V}\Big\|\frac{1}{|V|} - \mathcal{A}_{ij}^{t-s}\Big\|\|u_j^s\| \\ &+ \sum_{s=1}^{t}\sum_{j\in V}\Big\|\frac{1}{|V|} - \mathcal{A}_{ij}^{t-s+1}\Big\|\|w_j^{s-1}\| \end{aligned}\tag{25}$$
To further simplify the result, we bound the second term of (25) as follows
$$\begin{aligned}\|u_j^t\| &= \Big\|x_i^t - \sum_{j\in V} a_{ij}y_j^{t-1}\Big\| \\ &\leq \Big\|x_i^t - \sum_{j\in V} a_{ij}x_j^{t-1}\Big\| + \Big\|\sum_{j\in V} a_{ij}x_j^{t-1} - \sum_{j\in V} a_{ij}y_j^{t-1}\Big\| \\ &\leq \Big\|\sum_{j\in V} a_{ij}x_j^{t-1} - \sum_{j\in V} a_{ij}y_j^{t-1}\Big\| + \eta_t\|\mathrm{g}_i^{t-1}\| \\ &\leq \sum_{j\in V}\|w_j^{t-1}\| + \eta_t L_1 \end{aligned}\tag{26}$$
Substituting u_j^s with the result in (26) and rearranging three terms, we yield
$$\begin{aligned}&\mathbb{E}\|\bar{x}^t - x_i^t\| \\ &\leq \sum_{j\in V}\mathbb{E}\Big\|\frac{1}{|V|} - \mathcal{A}_{ij}^t\Big\|\|x_j^0\| \\ &\quad + \sum_{s=1}^{t}\sum_{j\in V}\mathbb{E}\Big\|\frac{1}{|V|} - \mathcal{A}_{ij}^{t-s}\Big\|\Big(\|w_j^{s-1}\| + \eta_s L_1\Big) \\ &\quad + \sum_{s=1}^{t}\sum_{j\in V}\mathbb{E}\Big\|\frac{1}{|V|} - \mathcal{A}_{ij}^{t-s+1}\Big\|\|w_j^{s-1}\| \\ &\overset{(a)}{\leq} \sum_{s=1}^{t}\sum_{j\in V}\mathbb{E}\Big[\gamma\lambda^{t-s}\Big(\|w_j^{s-1}\| + \eta_s L_1\Big) + \gamma\lambda^{t-s+1}\|w_j^{s-1}\|\Big] \\ &\overset{(b)}{\leq} \frac{\gamma|V|L_1[4\sqrt{2}m\varphi + \epsilon]}{\epsilon(1-\lambda)}\sum_{s=1}^{t}\eta_s, \end{aligned}\tag{27}$$
where (a) follows from Proposition 3 and assumption 3), and (b) holds since $\sum_{i\in V}\mathbb{E}\|w_i^t\| = \varphi|V|\sqrt{2m}\sigma_t \leq \frac{2\sqrt{2}m\varphi|V|L_1\eta_t}{\epsilon}$ according to Proposition 1 and Lemma 1. Then, we complete the proof. □

Proof of Theorem 2. We start proving Theorem 1 by observing the error between the local parameter of a device at time $t+1$ and the optimal solution x^* and have the following lemma

Lemma 2. By the assumptions, the following inequality holds
$$\begin{aligned}&\sum_{i\in V}\mathbb{E}\|x_i^{t+1} - x^*\|^2 \\ &\leq (1-\eta_t L_2)\sum_{i\in V}\mathbb{E}\|x_i^t - x^*\|^2 + \sum_{i\in V}\mathbb{E}\|w_i^t\|^2 \\ &\quad + 2\eta_t L_1\sum_{i\in V}\mathbb{E}\|w_i^t\| + 2\eta_t L_1|V|(\mathcal{B}_{avg}+2)\sum_{i\in V}\mathbb{E}\|x_i^t - \bar{x}^t\| \end{aligned}$$

$$+ |V|(L_1)^2(\eta_t)^2 + \eta_t L_2 \sum_{i\in V}\mathbb{E}\|\bar{x}^t - x_i^t\|^2 + 2\eta_t \sum_{i\in V}\mathbb{E}[f_i(x^*) - f_i(x_i^t)], \tag{28}$$

where $\mathcal{B}_{avg} = \mathbb{E}[\mathcal{B}_i^t]$.

Proof of Lemma 2.

$$\begin{aligned}\|x_i^{t+1} - x^*\|^2 &\overset{(a)}{\leq} \|\sum_{j\in V} a_{ij}y_j^t - \eta_t \mathrm{g}_i^t - x^*\|^2 \\ &\leq \sum_{j\in V} a_{ij}\|x_j^t - x^*\|^2 + \sum_{j\in V} a_{ij}\|w_j^t\|^2 \\ &\quad + 2\big\langle \sum_{j\in V} a_{ij}x_j^t - x^*, \sum_{j\in V} a_{ij}w_j^t\big\rangle \\ &\quad + (\eta_t)^2\|\mathrm{g}_i^t\|^2 - 2\big\langle \eta_t \mathrm{g}_i^t, \sum_{j\in V} a_{ij}y_j^t - x^*\big\rangle\end{aligned} \tag{29}$$

where (a) follows from the non-expensive property of the projection operator $\mathcal{P}_{\mathcal{X}}$ (i.e., $\|\mathcal{P}_{\mathcal{X}}(x_i^{t+1}) - \mathcal{P}_{\mathcal{X}}(x^*)\| \leq x_i^{t+1} - x^*$). The second-last term of (29) can be rewritten as

$$(\eta_t)^2\|\mathrm{g}_i^t\|^2 \overset{(a)}{\leq} (L_1)^2(\eta_t)^2, \tag{30}$$

where (a) follows from (20). Next, the last term of (29) can be rearranged as

$$\begin{aligned}&- 2\big\langle \eta_t \mathrm{g}_i^t, \sum_{j\in V} a_{ij}y_j^t - x^*\big\rangle \\ &\leq 2\big\|\eta_t \mathrm{g}_i^t\big\|\big\|\sum_{j\in V} a_{ij}y_j^t - \bar{x}^t\big\| - 2\big\langle \eta_t \mathrm{g}_i^t, \bar{x}^t - x^*\big\rangle \\ &\leq 2\eta_t L_1 \sum_{j\in V} a_{ij}\|w_j^t\| + 2\eta_t L_1 \sum_{j\in V} a_{ij}\|x_j^t - \bar{x}^t\| \\ &\quad - 2\eta_t\big\langle \mathrm{g}_i^t, \bar{x}^t - x_i^t\big\rangle + 2\eta_t\big\langle \mathrm{g}_i^t, x^* - x_i^t\big\rangle \\ &\overset{(a)}{\leq} 2\eta_t L_1 \sum_{j\in V} a_{ij}\|w_j^t\| + 2\eta_t L_1 \sum_{j\in V} a_{ij}\|x_j^t - \bar{x}^t\| \\ &\quad + 2\eta_t\Big[f_i(x_i^t) - f_i(\bar{x}^t) + \frac{L_2}{2}\|\bar{x}^t - x_i^t\|^2\Big] \\ &\quad + 2\eta_t\Big[f_i(x^*) - f_i(x_i^t) - \frac{L_2}{2}\|x^* - x_i^t\|^2\Big],\end{aligned} \tag{31}$$

where (a) follows from (21). Then, we bound the second last term of (31)

$$\begin{aligned}f_i(x_i^t) - f_i(\bar{x}^t) &= \sum_{s=1}^{\mathcal{B}_i^t}\big(f(x_i^t, \xi_{i,t}^s) - f(\bar{x}^t, \xi_{i,t}^s)\big) \\ &\overset{(a)}{\leq} \mathcal{B}_i^t\langle \mathrm{g}_i^t, x_i^t - \bar{x}^t\rangle \\ &\leq \mathcal{B}_i^t L_1\|x_i^t - \bar{x}^t\|\end{aligned} \tag{32}$$

where (a) follows from the property of sub-gradient. Combining (31) and (32), we obtain

$$\begin{aligned}&- 2\big\langle \eta_t \mathrm{g}_i^t, \sum_{j\in V} a_{ij}y_j^t - x^*\big\rangle \\ &\leq 2\eta_t L_1 \sum_{j\in V} a_{ij}\|w_j^t\| + 2\eta_t L_1 \sum_{j\in V} a_{ij}\|x_j^t - \bar{x}^t\| \\ &\quad + 2\eta_t \mathcal{B}_i^t L_1\|x_i^t - \bar{x}^t\| + \eta_t L_2\|\bar{x}^t - x_i^t\|^2 \\ &\quad + 2\eta_t\Big[f_i(x^*) - f_i(x_i^t) - \frac{L_2}{2}\|x^* - x_i^t\|^2\Big]\end{aligned} \tag{33}$$

Aftermath, substituting the second-last term and last term of (29) with (30) and (33), we obtain

$$\begin{aligned}\|x_i^{t+1} - x^*\|^2 &\leq \sum_{j\in V} a_{ij}\|x_j^t - x^*\|^2 + \sum_{j\in V} a_{ij}\|w_j^t\|^2 \\ &\quad + 2\big\langle \sum_{j\in V} a_{ij}x_j^t - x^*, \sum_{j\in V} a_{ij}w_j^t\big\rangle \\ &\quad + (L_1)^2(\eta_t)^2 + 2\eta_t L_1 \sum_{j\in V} a_{ij}\|w_j^t\| \\ &\quad + 2\eta_t L_1 \sum_{j\in V} a_{ij}\|x_j^t - \bar{x}^t\| \\ &\quad + 2\eta_t \mathcal{B}_i^t L_1\|x_i^t - \bar{x}^t\| + \eta_t L_2\|\bar{x}^t - x_i^t\|^2 \\ &\quad + 2\eta_t\Big[f_i(x^*) - f_i(x_i^t) - \frac{L_2}{2}\|x^* - x_i^t\|^2\Big]\end{aligned}$$

Taking expectation and summation over all the participants in the inequality above , we yield

$$\begin{aligned}&\sum_{i\in V}\mathbb{E}\|x_i^{t+1} - x^*\|^2 \\ &\overset{(a)}{\leq} (1 - \eta_t L_2)\sum_{i\in V}\mathbb{E}\|x_i^t - x^*\|^2 + \sum_{i\in V}\mathbb{E}\|w_i^t\|^2 \\ &\quad + 2\eta_t L_1 \sum_{i\in V}\mathbb{E}\|w_i^t\| + 2\eta_t L_1(|V|\mathcal{B}_{avg} + 1)\sum_{i\in V}\mathbb{E}\|x_i^t - \bar{x}^t\| \\ &\quad + |V|(L_1)^2(\eta_t)^2 + \eta_t L_2 \sum_{i\in V}\mathbb{E}\|\bar{x}^t - x_i^t\|^2 \\ &\quad + 2\eta_t \sum_{i\in V}\mathbb{E}[f_i(x^*) - f_i(x_i^t)],\end{aligned} \tag{34}$$

where (a) follows from the fact that $\mathbb{E}[w_j^t] = 0$ and the doubly stochastic matrix $\mathcal{A}$. The proof is completed. □

Now, we are ready to proceed with Theorem 2 by Lemma 2, and Theorem 1. By rearranging (28) and dividing both sides by $2\eta_t$, we obtain

$$\begin{aligned}&\sum_{i\in V}\mathbb{E}[f_i(x_i^t) - f_i(x^*)] \\ &\leq \frac{(1 - \eta_t L_2)}{2\eta_t}\sum_{i\in V}\mathbb{E}\|x_i^t - x^*\|^2 - \frac{1}{2\eta_t}\mathbb{E}\sum_{i\in V}\|x_i^{t+1} - x^*\|^2 \\ &\quad + \frac{L_2}{2}\sum_{i\in V}\mathbb{E}\|\bar{x}^t - x_i^t\|^2 + L_1(|V|\mathcal{B}_{avg} + 1)\sum_{i\in V}\mathbb{E}\|x_i^t - \bar{x}^t\| \\ &\quad + \frac{1}{2\eta_t}\sum_{i\in V}\mathbb{E}\|w_i^t\|^2 + L_1\sum_{i\in V}\|w_i^t\| + \frac{|V|(L_1)^2\eta_t}{2}\end{aligned} \tag{35}$$

It is obvious that

$$\begin{aligned}&f_i(x_j^t) - f_i(x^*) = f_i(x_j^t) - f_i(x_i^t) + f_i(x_i^t) - f_i(x^*) \text{ and} \\ &f_i(x_j^t) - f_i(x_i^t) \leq L_1\mathcal{B}_i^t(\|x_j^t - \bar{x}_t\| + \|x_i^t - \bar{x}_t\|)\end{aligned} \tag{36}$$

Combining (36) and applying Theorem 1 to (35), we obtain

$$\sum_{i\in V}\mathbb{E}[f_i(x_j^t) - f_i(x^*)]$$

$$
\begin{aligned}
&\leq \frac{(1-\eta_t L_2)}{2\eta_t}\sum_{i\in V}\mathbb{E}\|x_i^t - x^*\|^2 - \frac{1}{2\eta_t}\mathbb{E}\sum_{i\in V}\|x_i^{t+1} - x^*\|^2 \\
&\quad + \frac{L_2|V|}{2}\Big(\frac{\gamma|V|L_1[4\sqrt{2}m\varphi+\epsilon]}{\epsilon(1-\lambda)}\Big)^2\sum_{s=1}^{t}\eta_s \\
&\quad + ((|V|+2)\mathcal{B}_{avg}+1)\frac{\gamma|V|(L_1)^2[4\sqrt{2}m\varphi+\epsilon]}{\epsilon(1-\lambda)}\sum_{s=1}^{t}\eta_s \\
&\quad + \frac{1}{2\eta_t}\sum_{i\in V}\mathbb{E}\|w_i^t\|^2 + L_1\sum_{i\in V}\|w_i^t\| + \frac{|V|(L_1)^2\eta_t}{2}. \qquad (37)
\end{aligned}
$$

Let $\rho = L_1((|V|+2)T\mathcal{B}_{avg}+1)$. Summing over T rounds of (37), we can change the left hand side of (37) obtain

$$
\begin{aligned}
&\sum_{t\in T}\mathbb{E}[f_t(x_j^t) - f_t(x^*)] \\
&\leq \sum_{t\in T}\sum_{i\in V}\Big[\frac{(1-\eta_t L_2)}{2\eta_t}\mathbb{E}\|x_i^t - x^*\|^2 - \frac{1}{2\eta_t}\mathbb{E}\|x_i^{t+1} - x^*\|^2\Big] \\
&\quad + \frac{L_2|V|}{2}\Big(\frac{\gamma|V|L_1[4\sqrt{2}m\varphi+\epsilon]}{\epsilon(1-\lambda)}\Big)^2\sum_{t\in T}\eta_t \\
&\quad + \frac{\rho\gamma|V|L_1[4\sqrt{2}m\varphi+\epsilon]}{\epsilon(1-\lambda)}\sum_{t\in T}\eta_t \\
&\quad + \sum_{t\in T}\sum_{i\in V}\Big[\frac{1}{2\eta_t}\mathbb{E}\|w_i^t\|^2 + L_1\|w_i^t\|\Big] + \frac{|V|(L_1)^2}{2}\sum_{t\in T}\eta_t
\end{aligned} \qquad (38)
$$

Then, we bound each term of (38) one by one. The first and second terms can be written as follows

$$
\begin{aligned}
&\sum_{t\in T}\sum_{i\in V}\Big[\frac{(1-\eta_t L_2)}{2\eta_t}\mathbb{E}\|x_i^t - x^*\|^2 - \frac{1}{2\eta_t}\mathbb{E}\|x_i^{t+1} - x^*\|^2\Big] \\
&= \sum_{t\in T}\sum_{i\in V}\frac{-\eta_t L_2}{2\eta_t}\mathbb{E}\|x_i^t - x^*\|^2 - \sum_{i\in V}\frac{1}{2\eta_{T+1}}\mathbb{E}\|x_i^{T+1} - x^*\|^2,
\end{aligned}
$$

so it can be omitted. Let $\beta = \frac{\gamma|V|L_1[4\sqrt{2}m\varphi+\epsilon]}{\epsilon(1-\lambda)}$. The third and fourth terms can be rewritten as

$$
\begin{aligned}
&\frac{L_2|V|}{2}\Big(\frac{\gamma|V|L_1[4\sqrt{2}m\varphi+\epsilon]}{\epsilon(1-\lambda)}\Big)^2\sum_{t\in T}\eta_t \\
&\quad + \frac{\rho\gamma|V|L_1[4\sqrt{2}m\varphi+\epsilon]}{\epsilon(1-\lambda)}\sum_{t\in T}\eta_t \\
&= \big(\frac{L_2|V|}{2}\beta^2 + \rho\beta\big)\sum_{t\in T}\eta_t. \qquad (39)
\end{aligned}
$$

The fifth term follows from the inequality $\sum_{i\in V}\mathbb{E}\|w_i^t\|^2 = \varphi|V|2m(\sigma_t)^2 = \frac{\varphi|V|2m\Delta_t^2}{\epsilon^2} \leq \frac{\varphi|V|8(\eta_t)^2m^2(L_1)^2}{\epsilon^2}$ so we have

$$
\sum_{t\in T}\sum_{i\in V}\frac{1}{2\eta_t}\mathbb{E}\|w_i^t\|^2 \leq \frac{\varphi|V|4m^2(L_1)^2}{\epsilon^2}\sum_{t\in T}\eta_t \qquad (40)
$$

The sixth term follows the same trick and we have

$$
L_1\sum_{t\in T}\sum_{i\in V}\|w_i^t\| \leq \frac{\beta L_1}{\epsilon}\sum_{t\in T}\eta_t \qquad (41)
$$

Combining (39) , (40), and (41), we can rewrite (38)

$$
\begin{aligned}
&\sum_{t\in T}\mathbb{E}[f_t(x_j^t) - f_t(x^*)] \\
&\leq \big(\frac{L_2|V|}{2}\beta^2 + \rho\beta + \frac{\varphi|V|4m^2(L_1)^2}{\epsilon^2} + \frac{\beta L_1}{\epsilon}\big)\sum_{t\in T}\eta_t \qquad (42)
\end{aligned}
$$

where $\alpha = \beta + L_1|V|$. Finally, by setting $\eta_t \in \mathcal{O}(\frac{1}{\sqrt{T}}\frac{1}{(\mathcal{B}_{avg})})$ and $\epsilon \geq 1$, we can obtain

$$
\sum_{t\in T}\mathbb{E}[f_t(x_j^t) - f_t(x^*)] \leq \mathcal{O}\Big(\sqrt{T}\Big(\frac{\varphi^2}{\epsilon^2(1-\lambda)^2\mathcal{B}_{avg}}\Big)\Big) \qquad (43)
$$

Thus, the theorem follows. □

B. Hitting Time Table

TABLE IV
ENTRIES m_{ij} IN HIT. TIME MATRIX $\mathcal{M}(G_c)$ OF $\mathcal{H}$OA

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

C. Pseudocode of ODLF-PDP

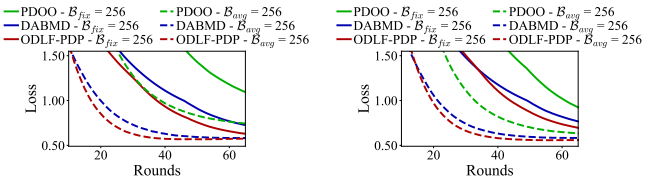
Algorithm 1 ODLF-PDP

Input: The initial model parameters x_i^0 for each devices $i \in V$, mini-batch size $\mathcal{B}$, communication topology $G_c = \{V, E_c\}$, consensus matrix $\mathcal{A}(G_c)$, time-varying step size η_t, privacy budget ϵ.

for all $t = 0, ..., T-1$ **do**
  for all device $i \in V$ **do** *in parallel*
    if DP mechanism is on **then**
      $\sigma_t \leftarrow \frac{2\eta^t\|g_i^{t-1}\|_1}{\epsilon}$;
      $w_i^t \leftarrow \mathcal{L}(0, 2(\sigma_t)^2)$;
      $y_i^t \leftarrow x_i^t + w_i^t$;
    else
      $y_i^t \leftarrow x_i^t$;
    Transmit y_i^t to neighbors and receive y_j^t from neighbors if $(i,j) \in E_c$
    for all $b_i^t \leq \mathcal{B}_i^t$ **do**
      $g_i^t \leftarrow g_i^t + \nabla f(x_i^t, \xi_i^t)$
      $b_i^t \leftarrow b_i^t + 1$
    end for
    $g_i^t \leftarrow g_i^t / b_i$
    $x_i^{t+1} \leftarrow \mathcal{P}_{\mathcal{X}}(\sum_{j=1}^n a_{ij}y_j^t - \alpha_t g_i^t)$
  end for
end for

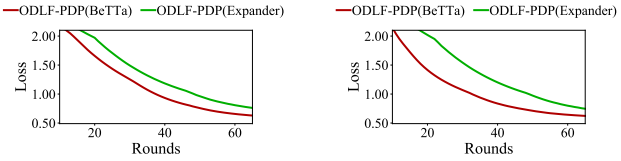
D. Additional Experiments

In Section IV only the results regarding *accuracy* are shown. Therefore, we appended the results regarding to *loss* in this section.



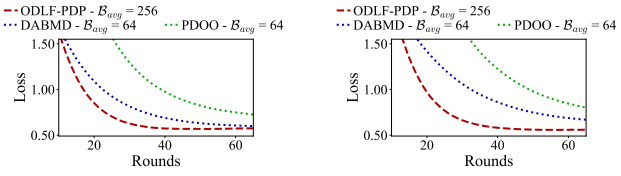
(a) Loss of $|V| = 16$ (b) Loss of $|V| = 32$

Fig. 7. Performance of three methods for different numbers of devices, the same fixed and average mini-batch sizes.



(a) Loss of $|V| = 16$ (b) Loss of $|V| = 32$

Fig. 8. Performance of ODLF-PDP(BeTTa) and ODLF-PDP(Expander) for different numbers of devices but the same mini-batch size.



(a) Loss of $|V| = 16$ (b) Loss of $|V| = 32$

Fig. 9. Effect of different numbers of devices. Note that the average mini-batch size $\mathcal{B}_{avg}$ is determined according to the network status by each method.