

# Low-latency Federated Learning with DNN Partition in Distributed Industrial IoT Networks

Xiumei Deng, *Student Member, IEEE*, Jun Li, *Senior Member, IEEE*, Chuan Ma, *Member, IEEE*, Kang Wei, *Graduate Student Member, IEEE*, Long Shi, *Senior Member, IEEE*, Ming Ding, *Senior Member, IEEE*, and Wen Chen, *Senior Member, IEEE*

**Abstract**—Federated Learning (FL) empowers Industrial Internet of Things (IIoT) with distributed intelligence of industrial automation thanks to its capability of distributed machine learning without any raw data exchange. However, it is rather challenging for lightweight IIoT devices to perform computation-intensive local model training over large-scale deep neural networks (DNNs). Driven by this issue, we develop a communication-computation efficient FL framework for resource-limited IIoT networks that integrates DNN partition technique into the standard FL mechanism, wherein IIoT devices perform local model training over the bottom layers of the objective DNN, and offload the top layers to the edge gateway side. Considering imbalanced data distribution, we derive the device-specific participation rate to involve the devices with better data distribution in more communication rounds. Upon deriving the device-specific participation rate, we propose to minimize the training delay under the constraints of device-specific participation rate, energy consumption and memory usage. To this end, we formulate a joint optimization problem of device scheduling and resource allocation (i.e. DNN partition point, channel assignment, transmit power, and computation frequency), and solve the long-term min-max mixed integer non-linear programming based on the Lyapunov technique. In particular, the proposed dynamic device scheduling and resource allocation (DDSRA) algorithm can achieve a trade-off to balance the training delay minimization and FL performance. We also provide the FL convergence bound for the DDSRA algorithm with both convex and non-convex settings. Experimental results demonstrate the derived device-specific participation rate in terms of feasibility, and show that the DDSRA algorithm outperforms baselines in terms of test accuracy and convergence time.

**Index Terms**—Federated learning, deep neural network (DNN) partition, device-specific participation rate, dynamic device scheduling and resource allocation.

## I. INTRODUCTION

RECENT advances in artificial intelligence (AI) and communication technologies along with wide deployment of the Industrial Internet of Things (IIoT) are leading us to Industry 4.0 [1], [2]. With the proliferation of modern

X. Deng, J. Li, K. Wei and L. Shi are with School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing, 210094, China. E-mail: {xiumeideng, jun.li, kang.wei}@njust.edu.cn, slong1007@gmail.com.

C. Ma is with Zhejiang Lab, Hangzhou, China. He is also with Nanjing University of Science and Technology, and Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education. E-mail: chuan.ma@zhejianglab.edu.cn.

M. Ding is with Data61, CSIRO, Sydney, NSW 2015, Australia. E-mail: ming.ding@data61.csiro.au.

W. Chen is with Department of Electronics Engineering, Shanghai Jiao Tong University, Shanghai 200240, China. E-mail: wenchen@sjtu.edu.cn.

Manuscript received May 16, 2022; revised September 06, 2022.

sensors and controllers, massive data has been collected and analyzed to derive intelligence, which transforms traditional industrial manufacturing to modernization and intelligence. Since raw industrial data often involves sensitive information, privacy is of the essence in industrial big data analysis. In traditional machine learning (ML), a large amount of raw data is collected from IIoT devices and centralized in a third party, which can lead to privacy leakage. Therefore, it is of crucial significance to process industrial raw data locally for the sake of privacy protection. As an emerging AI technology, federated learning (FL) has been regarded as a promising solution to privacy-preserving intelligent IIoT applications [3]. In a FL network, the centralized server aggregates the trained local ML models transmitted from the distributed devices. Compared with traditional ML, FL achieves a global ML model without any raw data exchange, thereby significantly reducing the communication overhead and promoting the privacy of each device. In this context, FL is widely applicable to a variety of IIoT scenarios wherein the local data samples possessed by every single device are insufficient to train an efficient ML model.

Wireless communication is of the essence in IIoT scenarios because it enables seamless, pervasive, and scalable connectivity among distributed devices without any cabled connections [4]. The total cost of deploying cabled communication can be relatively high in an industrial environment, especially when it requires equipment shutdown and pauses manufacturing lines. Therefore, many factories upgrade the existing equipment with industrial wireless components for additional intelligent applications (e.g., fault diagnosis and safety early warning), instead of deploying cables [5]. However, due to the limited communication resources (e.g., bandwidth), FL in IIoT systems suffers from inter-channel interference and prolonged transmission latency in the training process. Since IIoT systems demand the timely completion of each processing step during the manufacturing process, the communication overhead can be a bottleneck in FL-enabled IIoT systems. In addition, since IIoT devices are typically battery-operated, low-power consumption is vital to preserve battery life [6]. To this end, joint communication and energy resource allocation are of considerable significance to improve FL efficiency in wireless IIoT networks.

From the perspective of resource allocation, many recent studies have focused on how to achieve energy-efficient or communication-efficient FL. Authors in [7] derive the training loss gap between FL and centralized ML for a given duration

of communication rounds, and propose a device scheduling and resource allocation policy to enhance the FL performance by predicting channel state information. Based on deep multi-agent reinforcement learning, the works in [5] and [8] optimize the device selection as well as communication and computation resource allocation in an online manner to minimize FL loss function under delay and energy consumption constraints. To achieve a communication-efficient FL, [9]–[12] minimize the training latency by jointly optimizing communication and computation resource allocation as well as device selection. To evaluate the learning performance of the proposed low-latency FL framework, both FL training loss and learning delay are investigated to characterize the impact of device selection and resource allocation on the FL performance. In addition, to overcome the battery challenge of IIoT devices, the works in [13]–[17] propose different algorithms of joint communication and energy resource management, aiming at minimizing the total energy consumption of FL training process or weighted sum of energy consumption, latency and FL training loss.

Although emerging AI endows IIoT the capability to mine efficient knowledge from big data, the state-of-the-art deep neural network (DNN) architectures (e.g., GPT3) demand significant memory and computational resources [18]. Considering the huge computational cost of large-scale DNN training, the aforementioned works on communication and computation resource allocation are not adequate to reduce the computational burden on lightweight IIoT devices during the FL training process. To further reduce the computational cost of FL training for IIoT devices, recent works [19]–[21] on DNN partition assisted FL propose to divide the DNN model into two continuous portions, and separately train bottom and top layers of the DNN model at the device and edge server sides. However, these works focus on differentially private data perturbation mechanism designs to preserve the privacy of training data, and adopt predefined DNN partition strategies for all devices regardless of limited and heterogeneous computational resources. Later on, [22] and [23] propose to jointly optimize the partitioning and offloading of DNN inference tasks to reduce the total execution time. However, these works focus on DNN inference instead of DNN training. In fact, it is more challenging to optimize the DNN partition point in FL training process. This is due to the fact that FL demands the devices to synchronously perform the local model training in each communication round, while the DNN inference tasks can be independently executed by the devices. To our best knowledge, our paper is the first attempt to investigate the dynamic DNN partition in FL training process.

In addition, due to diverse computational capacity and memory resource among different IIoT devices, device heterogeneity introduces high training latency or even training failures, resulting in an unsatisfactory quality of experience (QoE) for real-time delay-sensitive applications. Furthermore, data heterogeneity can significantly degrade FL performance in the presence of non-independent and identically distributed (non-IID) data distribution [24]. To this end, proper participant device selection plays a crucial role in improving FL performance, especially when there exists a limit on the number of participant devices due to the limited communication

resources.

To address the aforementioned issues, we propose a communication-computation efficient FL framework for resource-limited IIoT networks that integrates the dynamic DNN partition technique into the standard FL mechanism. Considering limited and heterogeneous communication, energy, and memory resources as well as imbalanced data distribution, we propose a dynamic device scheduling and resource allocation policy to minimize the training latency while guaranteeing FL performance. Our contributions are summarized as follows:

- 1) By integrating DNN partition with FL, we propose a two-tier FL framework for IIoT networks wherein the devices hold the private datasets, perform the local model training over the bottom layers of the objective DNN, and offload the top layers to the edge gateway side at the middle tier. The roles of edge gateways include local model training of top DNN portion, device-level model combination, shop-floor-scale model aggregation, and model transmission. The base station (BS) at the top tier performs global model aggregation, and transfers scheduling policy information.
- 2) According to the forward and backward propagation, we derive the universal formulas for evaluating the layer-level memory usage and floating-point operation counts (FLOPs) based on the hyper-parameters of the DNN structure (e.g., filter size in convolution layer). As such, we propose a layer-level calculation model for training delay, memory usage and energy consumption in our two-tier FL framework.
- 3) We derive a divergence bound to analyze the impact of local dataset size and data distribution on FL training performance, and then develop the device-specific participation rate linked to the model performance. To achieve a low-latency FL, we formulate a joint dynamic optimization problem of the device scheduling and resource allocation (i.e., channel assignment, DNN partition point, transmit power and computation frequency) under the constraints of device-specific participation rate, energy consumption and memory usage. The objective of this optimization problem is to minimize the training latency while guaranteeing the learning performance of FL. To solve this long-term min-max mixed integer non-linear programming (MINLP) problem, we propose a dynamic device scheduling and resource allocation (DDSRA) algorithm to transform the stochastic optimization problem with a time-average device-specific participation rate constraint into a deterministic min-max MINLP problem based on the Lyapunov technique. Then, we solve the deterministic problem by the block coordinate descent method and bisection method in each communication round.
- 4) We conduct a performance analysis of the proposed DDSRA algorithm to verify its asymptotic optimality. A trade-off of  $[\mathcal{O}(1/V), \mathcal{O}(\sqrt{V})]$  is characterized between the FL training latency minimization and the degree of which the participation rate constraint is satisfied

with a control parameter  $V$ . This trade-off indicates that the minimization of the training latency and FL performance can be balanced by adjusting  $V$ . Furthermore, we provide the FL convergence bound for the DDSRA algorithm with both convex and non-convex settings. Our developed bound reveals that the FL convergence rate can be improved by increasing the training data size and setting a higher participation rate for the important devices with better data distribution.

- 5) Experimental results are provided to demonstrate the derived device-specific participation rate in terms of feasibility. Moreover, we analyze the participation rate of each device under the proposed DDSRA algorithm, and the experimental results show that the DDSRA algorithm outperforms the baselines in terms of learning accuracy and convergence time.

The remainder of this paper is organized as follows. In Section II, we briefly introduce the basic knowledge of FL and DNN partition technique. In Section III, we give the communication-computation efficient FL-enabled IIoT framework and then formulate the stochastic optimization problem. Section IV derives the device-specific participation rate, and Section V proposes the DDSRA algorithm. Section VI produces the performance analysis of the proposed algorithm to verify asymptotic optimality, and studies the FL convergence rate. Then, the experimental results are presented in Section VII. Section VIII concludes this paper. For ease of reference, Tables I lists the main notations used in this paper.

## II. PRELIMINARIES

### A. Federated Learning

FL enables local model training across distributed devices, and global model aggregation at a centralized server. Considering an FL network of  $N$  devices collaborating to train an ML model over their respective local datasets. The goal of FL is to find a set of model parameters  $\mathbf{w}$  that minimizes the global loss function  $F(\mathbf{w})$  on all the local datasets, i.e.,  $F(\mathbf{w}) = \frac{\sum_n |\mathcal{D}_n| F_n(\mathbf{w})}{\sum_n |\mathcal{D}_n|}$ , where  $F_n(\mathbf{w}) = f(\mathbf{w}, \mathcal{D}_n)$  denotes the loss function on the local dataset  $\mathcal{D}_n$ . To keep the training data localized and private, each device in FL utilizes the gradient-descent method to minimize the local loss function  $F_n(\mathbf{w})$  over its local dataset by iteratively moving in the negative direction of the gradient. To obtain the global model, they synchronously upload the trained local model parameters to the centralized server, which aggregates all the collected local model parameters and returns the result to each device to update the local model parameters. Compared with traditional ML, FL can collaboratively build a shared model without raw data exchange, which greatly reduces the communication overhead and promotes the privacy of localized data.

### B. Deep Neural Network Partition

1) *Deep Neural Network*: A deep neural network (DNN) can be considered as stacked layers of neural networks, where raw data gets passed to the input layer and the output layer outputs the prediction result. Each hidden layer takes in the

TABLE I: List of main notations.

Notations	Descriptions
$\mathcal{N}$	Index set of the end devices
$\mathcal{M}$	Index set of the edge gateways
$\mathbf{a}$	Deployment matrix
$\mathcal{D}_n$	Local dataset
$\mathcal{L}$	Index set of the DNN layers
$\mathcal{J}$	Index set of the available channels
$l_n(t)$	DNN partition point
$\mathbf{I}(t)$	Channel assignment matrix
$K$	Local iterations
$\beta$	Step size
$\bar{D}_n$	Number of sample points
$o_l$	FLOPs of the forward propagation for each sample point in the $l$ -th layer
$o'_l$	FLOPs of the backward propagation for each sample point in the $l$ -th layer
$\phi_n^D$	FLOPs per clock cycle of the $n$ -th device
$\phi_m^G$	FLOPs per clock cycle of the $m$ -th gateway
$f_n^D$	Computation frequency of the $n$ -th device
$f_{m,n}^G(t)$	Computation frequency of the $m$ -th gateway assigned to the local model training offloaded from the $n$ -th device in the $t$ -th communication round
$e_n^{\text{tra,D}}(t)$	Energy consumption of the $n$ -th device for local model training in the $t$ -th communication round
$v_n^D$	Effective switched capacitance of the $n$ -th device
$e_m^{\text{tra,G}}(t)$	Energy consumption of the $m$ -th gateway for local model training in the $t$ -th communication round
$g_{n,l}$	Memory usage of the $l$ -th layer for storing the model parameters and intermediate data in the forward and backward propagation
$G_n^D(t)$	Memory usage for the bottom DNN layers trained at the $n$ -th device in the $t$ -th communication round
$G_m^G(t)$	Memory usage for the top DNN layers trained at the $m$ -th gateway in the $t$ -th communication round
$G_n^{\text{D,max}}$	Memory size of the $n$ -th device
$G_m^{\text{G,max}}$	Memory size of the $m$ -th gateway
$h_{m,j}^d(t)$	Downlink channel power gain from the BS to the $m$ -th gateway via the $j$ -th channel in the $t$ -th communication round
$B^d$	Bandwidth of the downlink channel
$P^B$	Transmit power of the BS
$N_0$	Noise power spectral density
$\gamma$	DNN model size
$i_{m,j}^d(t)$	Co-channel interference of the downlink channel
$B^u$	Bandwidth of the uplink channel
$P_m(t)$	Transmit power of the $m$ -th gateway in the $t$ -th communication round
$h_{m,j}^u(t)$	Uplink channel power gain from the $m$ -th gateway to the BS in the $t$ -th communication round
$i_{m,j}^u(t)$	Co-channel interference of the uplink channel
$e_m^{\text{up}}(t)$	Energy consumption of the $m$ -th gateway for model transmitting in the $t$ -th communication round
$E_n^D(t)$	Energy arrival at the $n$ -th device in the $t$ -th communication round
$E_m^G(t)$	Energy arrival at the $m$ -th gateway in the $t$ -th communication round
$e_m^G(t)$	Total energy consumption of the $m$ -th gateway in the $t$ -th communication round
$\tau(t)$	Total latency of the $t$ -th communication round
$\Gamma_m$	Participation rate of the $m$ -th gateway and its associated devices

inputs, passes the weighted inputs into an activation function along with the biases, and forwards the outputs to the next layer.

2) *Forward and Backward Propagation*: According to the gradient-descent method, the backpropagation algorithm is utilized to calculate the gradient of the objective loss function with respect to the model parameters (i.e., neural network's

weights and biases) by the chain rule [25]. Specifically, forward and backward propagation are executed in each iteration until the objective loss function converges. In the forward propagation stage, each hidden layer calculates the outputs by adding the biases to the weighted inputs and passing results into the activation function. The data flows from the first input layer to the last output layer to obtain the prediction result, where the output of each hidden layer serves as the input of the next one. In the backward propagation stage, error propagates in the opposite direction from the output layer to the input layer in order to compute the gradient of the loss function with regard to the model parameters. The error term in the output layer is calculated as the difference of actual and desired output, and the error term in each hidden layer is calculated as the weighted sum of the errors from the next layer. Based on the error passed from the next layer, each layer computes the gradient to update the model parameters and the error to be propagated to the previous layer.

3) *Deep Neural Network Partition*: In DNN partition mechanism, we set a partition point to divide the objective DNN into two continuous portions, and separately deploy the bottom and top layers of the DNN at an end device and an edge server. To perform the forward and backward propagation, the end device first transmits the labels of training dataset to the edge server. Then, the output of the last layer in the device-side DNN is transmitted to the edge server during the forward propagation stage, while the error term of first layer in the server-side DNN is transmitted to the end device during the backward propagation stage. Based on DNN partition mechanism, the local model training of top DNN portion is offloaded to the edge server, thereby greatly reducing the computational burden on the resource-constrained device side. Upon completing the DNN model training, we perform the model combination, i.e., combining the bottom and top layers of the trained DNN model to obtain the complete DNN model. The decision of DNN partition point mainly depends on three aspects as follows: (a) computational resources (i.e., processing power, memory capacity, etc.) of edge server. The computational resources required by the training of the offloaded DNN layers cannot exceed the computational resources of the edge server; (b) communication overhead. In the training process, DNNs can be partitioned in pooling layers to reduce the data size of forward outputs and errors transmitted between the end device and edge server; (c) privacy concern. Deeper DNN partition points can help mitigating the potential threats of privacy leakage.

### III. SYSTEM MODEL

#### A. System Overview

As shown in Fig.1, we consider an FL-enabled IIoT network with  $M$  shop floors, wherein each shop floor employs a single edge gateway and a group of end devices. In every shop floor, each device monitors the manufacturing process, collects local datasets, and performs local model training. Due to limited computation and memory resources at the device side, each device trains bottom layers of the objective DNN locally, and the training of top layers is offloaded to the edge gateway in the

same shop floor. Then, the gateway collects the local models, combines the bottom and top DNN portions, and performs the shop-floor-scale model aggregation. The base station (BS) collects the aggregated shop-floor-scale models, and performs the global model aggregation to obtain a shared model.

1) *End devices*: Let  $\mathcal{N} = \{1, \dots, N\}$  and  $\mathcal{M} = \{1, \dots, M\}$  denote the index sets of the devices and gateways, respectively. Define an  $N \times M$  deployment matrix as  $\mathbf{a}$  with entry  $a_{n,m} \in \{0, 1\}$ ,  $n \in \mathcal{N}$  and  $m \in \mathcal{M}$ . If  $a_{n,m} = 1$ , the  $n$ -th device is deployed with the  $m$ -th gateway in the  $m$ -th shop floor. As such, the deployment matrix satisfies  $\sum_{n \in \mathcal{N}} a_{n,m} = 1, \forall m \in \mathcal{M}$ . Each group of devices can only communicate with the gateway in the same shop floor, and we describe the group of devices as the associated devices with the gateway in the same shop floor. Each device holds a local dataset  $\mathcal{D}_n = \{\mathbf{x}_{n,i} \in \mathbb{R}^d, y_{n,i} \in \mathbb{R}\}_{i=1}^{D_n}$  with  $D_n = |\mathcal{D}_n|$  data points, where  $\mathbf{x}_{n,i}$  and  $y_{n,i}$  are the feature vector and label for the  $i$ -th data point at the  $n$ -th device. Let  $\mathcal{L} = \{1, \dots, L\}$  denote the index set of the DNN layers. For local model training, the bottom  $l_n(t)$  layers of the objective DNN are trained locally at the  $n$ -th device in the  $t$ -th communication round, while the training of top  $L - l_n(t)$  layers are offloaded to the associated gateway.

2) *Edge gateways*: The roles of each gateway include local model training of top DNN portion, device-level model combination, shop-floor-scale model aggregation, and model transmission. First, each gateway performs the forward and backward propagation for the top layers of the objective DNNs offloaded from the associated devices. Second, each gateway collects the bottom layers of the DNNs from the associated devices, combines the bottom and top layers of the trained DNNs, aggregates the combined local models, and transmits the aggregated shop-floor-scale model to the BS. Note that only a part of the shop floors can be selected to participate in FL in each communication round.

3) *Base station*: Let  $\mathcal{J} = \{1, \dots, J\}$  denote the index set of the available channels,  $\mathcal{T} = \{1, \dots, T\}$  denote the index set of communication rounds. Orthogonal frequency-division multiplexing (OFDM) is adopted to transmit the shop-floor-scale model parameters from the gateways to the BS in parallel. In each communication round,  $J$  selected gateways can communicate with the BS through the assigned channels. The BS equipped with a cloud server has two functions: (a) aggregating the model parameters received from the selected edge gateways; (b) sending back the global model parameters and the scheduling policy information (e.g., channel assignment) to the gateways.

Consider that FL operation in each communication round is synchronous. As Fig.1 illustrated, the FL in the  $t$ -th communication round operates in the following steps:

- 1) At the beginning of the  $t$ -th communication round, the BS selects  $J$  gateways according to scheduling policy, and broadcasts the global model parameters  $\mathbf{W}^t$  to the selected gateways. Define the channel assignment matrix as  $\mathbf{I}(t)$  with entry  $I_{m,j}(t) \in \{0, 1\}$ ,  $m \in \mathcal{M}$  and  $j \in \mathcal{J}$ . If  $I_{m,j}(t) = 1$ , the  $m$ -th gateway is assigned to the  $j$ -th channel in the  $t$ -th communication round. Note that the channel assignment matrix satisfies  $\sum_{m \in \mathcal{M}} I_{m,j}(t) = 1$

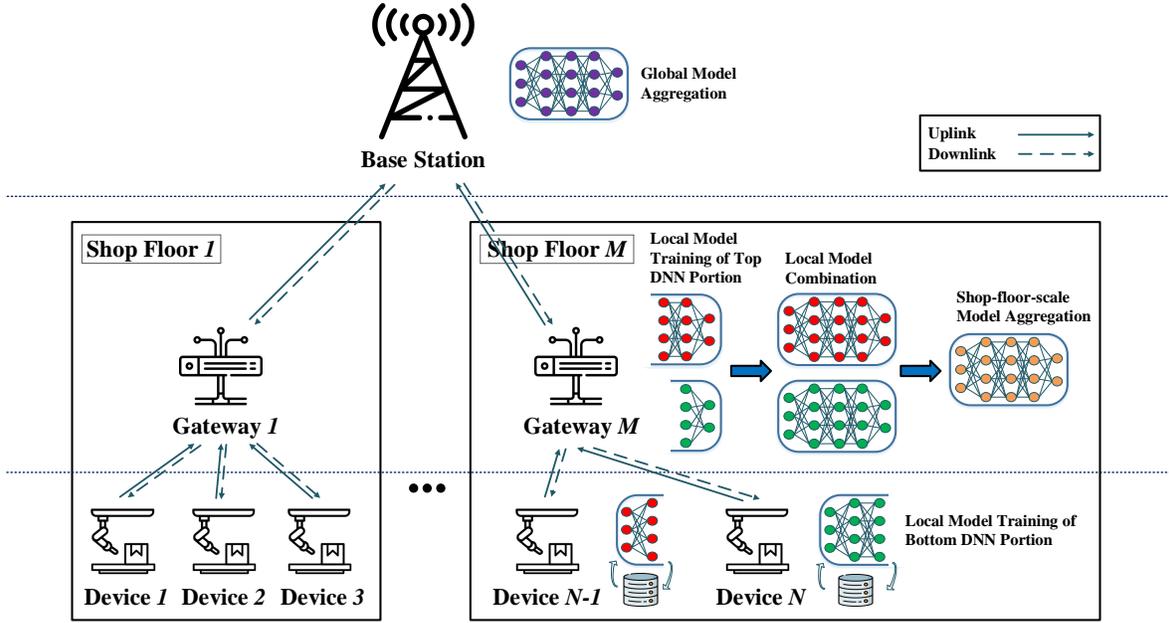


Fig. 1: System model of a two-tier communication-computation efficient FL-enabled IIoT framework.

and  $\sum_{j \in \mathcal{J}} I_{m,j}(t) \leq 1, \forall t \in \mathcal{T}$ . Then, each selected gateway broadcasts the global model parameters  $\mathbf{W}^t$  to its associated devices.

- 2) Upon receiving  $\mathbf{W}^t$ , each training device and the associated gateway collaboratively perform the forward and backward propagation by DNN partition mechanism to update the local model parameters. Let  $\tilde{\mathbf{w}}_n^{0,t} = \mathbf{W}^t$  denote the initial local model parameters of the  $n$ -th device in the  $t$ -th communication round. For each training device and the associated gateway, local model parameters are updated according to the gradient-descent update rule with respect to the local loss function over a total of  $K$  iterations. The update rule in the  $k$ -th iteration is  $\tilde{\mathbf{w}}_n^{k,t} = \tilde{\mathbf{w}}_n^{k-1,t} - \beta \nabla \tilde{F}_n(\tilde{\mathbf{w}}_n^{k-1,t})$ , where  $\tilde{\mathbf{w}}_n^{k,t}$  denotes the local model parameters of the  $n$ -th device in the  $k$ -th iteration and the  $t$ -th communication round,  $\beta > 0$  is the step size,  $\nabla \tilde{F}_n(\tilde{\mathbf{w}}_n^{k-1,t}) = \nabla f_n(\tilde{\mathbf{w}}_n^{k-1,t}, \tilde{\mathcal{D}}_n)$  is the stochastic gradient of local loss function, and  $\tilde{\mathcal{D}}_n$  is a batch of the local dataset  $\mathcal{D}_n$  with  $|\tilde{\mathcal{D}}_n|$  sample points.
- 3) Upon completing the local model training, each training device transmits the bottom layers of the DNN to the associated gateway. Then, the selected gateways combine the bottom and top layers of the trained DNNs, aggregate the combined local model parameters according to the federated averaging (FedAvg) algorithm [26], i.e.,  $\hat{\mathbf{w}}_m^t = \frac{\sum_{n \in \mathcal{N}} a_{n,m} \tilde{\mathcal{D}}_n \tilde{\mathbf{w}}_n^{K,t}}{\sum_{n \in \mathcal{N}} a_{n,m} \tilde{\mathcal{D}}_n}$ , and transmit the aggregated shop-floor-scale model parameters  $\hat{\mathbf{w}}_m^t$  to the BS. With the received model parameters uploaded from the selected gateways, the BS updates the global model parameters by performing global aggregation according to FedAvg, i.e.,  $\mathbf{W}^{t+1} = \frac{\sum_{m \in \mathcal{M}} \sum_{j \in \mathcal{J}} I_{m,j}(t) D_m \hat{\mathbf{w}}_m^t}{\sum_{m \in \mathcal{M}} \sum_{j \in \mathcal{J}} I_{m,j}(t) D_m}$ , where  $D_m = \sum_{n \in \mathcal{N}} a_{n,m} \tilde{\mathcal{D}}_n$ .

## B. Computation Model

Before delving into the computation model, we first define the following notations for the hyper-parameters and tensor shapes in the forward and backward propagation. Let  $B_s$  and  $S_f$  denote the batch size and the precision format of the data type, respectively. For the convolution layer and pooling layer, respectively;  $H_o$ ,  $W_o$  and  $C_o$  are output height, width, and channel, respectively;  $H_i$ ,  $W_i$  and  $C_i$  are input height, width, and channel;  $H_f$  and  $W_f$  are the filter's height and width. For the fully connected layer,  $S_i$  and  $S_o$  are the input and output sizes. To calculate the memory usage and FLOPs for the bottom and top DNN portions trained at the device and gateway side, we list the main layer-level memory usage and FLOPs in Table II according to the backpropagation algorithm [27], [28].

Let  $o_l$  and  $o'_l$  denote the FLOPs of the forward and backward propagation for each sample point in the  $l$ -th layer, respectively. As such, the local model training time of the  $m$ -th gateway and its associated devices in the  $t$ -th communication round is represented as

$$\tau_m^{\text{tra}}(t) = \sum_{j \in \mathcal{J}} I_{m,j}(t) \max_{n \in \mathcal{N}} \left\{ a_{m,n} K \tilde{\mathcal{D}}_n \left( \frac{\sum_{l=1}^{l_n(t)} (o_l + o'_l)}{\phi_n^D f_n^D} + \frac{\sum_{l=l_n(t)+1}^L (o_l + o'_l)}{\phi_m^G f_{m,n}^G(t)} \right) \right\}, \quad (1)$$

where  $\phi_n^D$  and  $\phi_m^G$  are the FLOPs per clock cycle of the  $n$ -th device and the  $m$ -th gateway,  $f_n^D$  is the computation frequency of the  $n$ -th device for local model training, and  $f_{m,n}^G(t)$  is the computation frequency of the  $m$ -th gateway assigned to the  $n$ -th device's local model training. Note that  $f_{m,n}^G(t)$  is limited by the total computation frequency of the  $m$ -th gateway, i.e.,  $\sum_{n \in \mathcal{N}} a_{m,n} f_{m,n}^G(t) \leq f_m^{\text{G,max}}$ . The energy consumption of the

TABLE II: Layer-level memory usage and FLOPs in DNN forward and backward propagation operations

Layer Category	Memory Usage		Floating-point Operation	
	Tensor Category	Tensor Size	Operator Category	FLOPs
Convolution	Weight	$S_f C_i H_f W_f C_o$	Forward Propagation	$2B_s C_i H_f W_f C_o H_o W_o$
	Forward Outout	$S_f B_s C_o H_o W_o$	Error Calculation	$2B_s(2W_f + W_f W_o - 2) \times (2H_f + H_f H_o - 2)$
	Backward Error	$S_f B_s C_i H_i W_i$		
	Gradient	$S_f C_i H_f W_f C_o$	Gradient Calculation	$2B_s C_i H_f W_f C_o H_o W_o$
Pooling	Forward Outout	$S_f B_s C_o H_o W_o$	Forward Propagation	$B_s C_i H_i W_i$
	Backward Error	$S_f B_s C_i H_i W_i$	Error Calculation	$B_s C_i H_i W_i$
Fully Connected	Weight	$S_i S_o$	Forward Propagation	$2B_s S_i S_o$
	Forward Outout	$B_s S_o$	Error Calculation	$2B_s S_i S_o$
	Backward Error	$B_s S_i$		
	Gradient	$S_i S_o$	Gradient Calculation	$B_s S_i S_o$

$n$ -th device for local model training in the  $t$ -th communication round can be expressed as [23]

$$e_n^{\text{tra,D}}(t) = \sum_{j \in \mathcal{J}} \sum_{m \in \mathcal{M}} I_{m,j}(t) a_{m,n} K \tilde{D}_n v_n^{\text{D}} / \phi_n^{\text{D}} \left( \sum_{l=1}^{l_n(t)} (o_l + o'_l) \right) (f_n^{\text{D}})^2, \quad (2)$$

where  $v_n^{\text{D}}$  is the effective switched capacitance. Moreover, the energy consumption of the  $m$ -th gateway for the local model training offloaded from its associated devices in the  $t$ -th communication round is given by

$$e_m^{\text{tra,G}}(t) = \sum_{j \in \mathcal{J}} \sum_{n \in \mathcal{N}} I_{m,j}(t) a_{m,n} K \tilde{D}_n v_m^{\text{G}} / \phi_m^{\text{G}} \left( \sum_{l=l_n(t)+1}^L (o_l + o'_l) \right) (f_{m,n}^{\text{G}}(t))^2. \quad (3)$$

For the  $n$ -th training device with the training dataset  $\tilde{D}_n$ , let  $g_{n,l}$  denote the memory usage of the  $l$ -th layer for storing the model parameters and intermediate data in the forward and backward propagation. The total memory usage for the bottom and top layers of the objective DNN, which are trained at the device and gateway side, are given by

$$G_n^{\text{D}}(t) = \sum_{j \in \mathcal{J}} \sum_{m \in \mathcal{M}} \sum_{l=1}^{l_n(t)} I_{m,j}(t) a_{m,n} g_{n,l}, \quad (4)$$

and

$$G_m^{\text{G}}(t) = \sum_{j \in \mathcal{J}} \sum_{n \in \mathcal{N}} \sum_{l=l_n(t)+1}^L I_{m,j}(t) a_{m,n} g_{n,l}. \quad (5)$$

Let  $G_n^{\text{D,max}}$  and  $G_m^{\text{G,max}}$  denote the memory size of the  $n$ -th device and the  $m$ -th gateway, respectively. We can note that  $0 \leq G_n^{\text{D}}(t) \leq G_n^{\text{D,max}}$  and  $0 \leq G_m^{\text{G}}(t) \leq G_m^{\text{G,max}}$ , since the memory usage cannot exceed the memory size of the equipment. In addition, the local model training cannot be fully offloaded due to the limited memory and energy resources at the edge gateway side.

### C. Communication Model

At the beginning of the  $t$ -th communication round, the BS broadcasts the global model parameters to the selected gateways over wireless channels. Assume that the wireless

channels are IID block fading. The channel remains static in each communication round but varies among different communication rounds. In our communication model, the downlink channel power gain from the BS to the  $m$ -th gateway via the  $j$ -th channel is modeled as  $h_{m,j}^{\text{d}}(t) = h_0 \rho_{m,j}^{\text{d}}(t) (d_0/d_m)^\nu$ , where  $h_0$  is the path loss constant,  $\rho_{m,j}^{\text{d}}(t)$  is the small-scale fading channel power gain from the BS to the  $m$ -th gateway via the  $j$ -th channel in the  $t$ -th communication round,  $d_m$  is the distance from the BS to the  $m$ -th gateway,  $d_0$  is the reference distance, and  $\nu$  is the large-scale path loss factor, respectively. Thus, the global model transmission time from the BS to the  $m$ -th gateway in the  $t$ -th communication round can be represented as

$$\tau_m^{\text{down}}(t) = \sum_{j \in \mathcal{J}} \frac{I_{m,j}(t) \gamma}{B^{\text{d}} \log_2 \left( 1 + \frac{P^{\text{B}} h_{m,j}^{\text{d}}(t)}{B^{\text{d}} N_0 + i_{m,j}^{\text{d}}(t)} \right)}, \quad (6)$$

where  $\gamma$  is DNN model size,  $B^{\text{d}}$  is the bandwidth of the downlink channel,  $P^{\text{B}}$  is the transmit power of the BS,  $N_0$  is the noise power spectral density, and  $i_{m,j}^{\text{d}}(t)$  is the co-channel interference caused by radio communication services in other areas, respectively.

After downloading the global model parameters from the BS, each selected gateway broadcasts the global model parameters to its associated devices. Due to the short-distance wireless technology, we consider that the transmission time between the gateways and the associated devices is negligible compared with the overall FL training delay [29]–[31]. After completing the local model training, each training device transmits the bottom layers of the DNN to its associated gateway, and the selected gateways perform the device-level model combination and forward the aggregated shop-floor-scale model parameters to the BS over wireless links. Similarly, the model transmission time from the  $m$ -th gateway to the BS in the  $t$ -th communication round is

$$\tau_m^{\text{up}}(t) = \sum_{j \in \mathcal{J}} \frac{I_{m,j}(t) \gamma}{B^{\text{u}} \log_2 \left( 1 + \frac{P_m(t) h_{m,j}^{\text{u}}(t)}{B^{\text{u}} N_0 + i_{m,j}^{\text{u}}(t)} \right)}, \quad (7)$$

where  $B^{\text{u}}$  represents the bandwidth of the uplink channel,  $P_m(t)$  denotes the transmit power of the  $m$ -th gateway,  $i_{m,j}^{\text{u}}(t)$  is the co-channel interference,  $h_{m,j}^{\text{u}}(t) = h_0 \rho_{m,j}^{\text{u}}(t) (d_0/d_m)^\nu$  is the uplink channel power gain from the  $m$ -th gateway to the BS, and  $\rho_{m,j}^{\text{u}}(t)$  is the small-scale fading channel power gain, respectively. The energy consumption of the  $m$ -th gateway for

transmitting the aggregated shop-floor-scale model parameters in the  $t$ -th communication round is

$$e_m^{\text{up}}(t) = \sum_{j \in \mathcal{J}} \frac{P_m(t) I_{m,j}(t) \gamma}{B^u \log_2 \left( 1 + \frac{P_m(t) h_{m,j}^u(t)}{B^u N_0 + i_{m,j}^u(t)} \right)}. \quad (8)$$

In addition, the energy harvesting (EH) components equipped at devices and gateways harvest renewable energy from the nature for local model training and transmission. We formulate the EH process as successive energy packet arrivals. Let  $E_n^{\text{D}}(t)$  and  $E_m^{\text{G}}(t)$  denote the energy arrival at the  $n$ -th device and the  $m$ -th gateway in the  $t$ -th communication round. Consider that  $E_n^{\text{D}}(t)$  and  $E_m^{\text{G}}(t)$  are modeled as IID stochastic processes, i.e.,  $E_n^{\text{D}}(t)$  and  $E_m^{\text{G}}(t)$  are uniformly distributed within  $[0, E_n^{\text{D,max}}]$  and  $[0, E_m^{\text{G,max}}]$ , respectively. Note that the total energy consumption of the  $m$ -th gateway in the  $t$ -th communication round can be represented as

$$e_m^{\text{G}}(t) = e_m^{\text{tra,G}}(t) + e_m^{\text{up}}(t). \quad (9)$$

As such, it can be derived that  $0 \leq e_n^{\text{tra,D}}(t) \leq E_n^{\text{D}}(t)$ , and  $0 \leq e_m^{\text{G}}(t) \leq E_m^{\text{G}}(t)$ , since the energy consumption cannot exceed the energy arrivals in each communication round.

#### D. Problem Formulation

According to the analysis above, the time consumption of each communication round mainly comes from three parts, i.e., global model downloading, local model training, and shop-floor-scale model uploading. Thus, the total delay of the  $t$ -th communication round is given by

$$\tau(t) = \max_{m \in \mathcal{M}} \{ \tau_m^{\text{tra}}(t) + \tau_m^{\text{up}}(t) + \tau_m^{\text{down}}(t) \}. \quad (10)$$

To obtain a communication-computation efficient FL framework, we develop a dynamic device selection and resource scheduling protocol to minimize average delay under the energy consumption and memory usage constraints. Let  $\mathbf{X}(t) = [\mathbf{I}(t), \mathbf{l}(t), \mathbf{P}(t), \mathbf{f}^{\text{G}}(t)]$ . In this context, we formulate a stochastic optimization problem as

$$\mathbf{P0} : \min_{\mathbf{X}(t)} \frac{1}{T} \sum_{t=1}^T \tau(t) \quad (11)$$

- s.t. **C1** :  $I_{m,j}(t) \in \{0, 1\}, \forall m \in \mathcal{M}, j \in \mathcal{J}, t \in \mathcal{T}$ ,  
**C2** :  $\sum_{j \in \mathcal{J}} I_{m,j}(t) \leq 1, \forall m \in \mathcal{M}, t \in \mathcal{T}$ ,  
**C3** :  $\sum_{m \in \mathcal{M}} I_{m,j}(t) = 1, \forall j \in \mathcal{J}, t \in \mathcal{T}$ ,  
**C4** :  $0 \leq P_m(t) \leq P_m^{\text{max}}, \forall m \in \mathcal{M}, t \in \mathcal{T}$ ,  
**C5** :  $0 \leq l_n(t) \leq L, \forall n \in \mathcal{N}, t \in \mathcal{T}$ ,  
**C6** :  $f_m^{\text{G,min}} \leq \sum_{n \in \mathcal{N}} a_{m,n} f_{m,n}^{\text{G}}(t) \leq f_m^{\text{G,max}}, \forall m \in \mathcal{M}, t \in \mathcal{T}$ ,  
**C7** :  $0 \leq G_n^{\text{D}}(t) \leq G_n^{\text{D,max}}, \forall n \in \mathcal{N}, t \in \mathcal{T}$ ,  
**C8** :  $0 \leq G_m^{\text{G}}(t) \leq G_m^{\text{G,max}}, \forall m \in \mathcal{M}, t \in \mathcal{T}$ ,  
**C9** :  $0 \leq e_n^{\text{tra,D}}(t) \leq E_n^{\text{D}}(t), \forall n \in \mathcal{N}, t \in \mathcal{T}$ ,  
**C10** :  $0 \leq e_m^{\text{G}}(t) \leq E_m^{\text{G}}(t), \forall m \in \mathcal{M}, t \in \mathcal{T}$ ,  
**C11** :  $\frac{1}{T} \sum_{t=1}^T \mathbb{1}_m^t \geq \Gamma_m, \forall m \in \mathcal{M}$ ,

where  $\mathbb{1}_m^t = \sum_{j \in \mathcal{J}} I_{m,j}(t)$  indicates whether the  $m$ -th gateway is selected to participate in the local model training in the  $t$ -th communication round. That is, if  $\mathbb{1}_m^t = 1$ , the  $m$ -th gateway and associated devices are selected to train the local model in the  $t$ -th communication round.  $\Gamma_m$  is the participation rate of the  $m$ -th gateway and its associated devices derived in the following section. The ranges of the variables  $\mathbf{I}(t)$ ,  $\mathbf{l}(t)$ ,  $\mathbf{P}(t)$  and  $\mathbf{f}^{\text{G}}(t)$  are constrained by **C1** ~ **C6**, respectively. **C7** ~ **C10** are the memory usage and energy consumption constraints for devices and gateways in each communication round, respectively. Furthermore, the long-term constraint **C11** is adopted to optimize the FL performance by guaranteeing the participation rate for each gateway and the associated devices. Overall, the goal of **P0** is to jointly optimize communication and computation resources under memory usage, energy consumption and participation rate constraints.

#### IV. DEVICE-SPECIFIC PARTICIPATE RATE

In this section, we derive a model divergence bound to measure the learning performance of each gateway and the associated devices' local model training. As such, the participation rate of each gateway and the associated devices can be determined based on the derived divergence bound. Our analysis of the model divergence bound focuses on three parts, i.e., data distribution, training dataset size, and the number of local epochs.

Before the analysis, two auxiliary notations are introduced. We use  $\mathbf{w}_n^{k,t}$  to denote the set of local model parameters that follows a full gradient descent, i.e.,  $\mathbf{w}_n^{k+1,t} = \mathbf{w}_n^{k,t} - \beta \nabla f(\mathbf{w}_n^{k,t}, \mathcal{D}_n)$ , and  $\mathbf{v}^{k,t}$  to denote the set of local model parameters that follows a centralized gradient descent, i.e.,  $\mathbf{v}^{k+1,t} = \mathbf{v}^{k,t} - \beta \nabla f(\mathbf{v}^{k,t}, \cup \mathcal{D}_n)$ . Note that although the sets of model parameters  $\tilde{\mathbf{w}}_n^{k,t}$ ,  $\mathbf{w}_n^{k,t}$ , and  $\mathbf{v}^{k,t}$  follow different update rules, they are synchronized with  $\tilde{\mathbf{w}}^{K,t-1}$  at the beginning of the  $t$ -th communication round, i.e.,  $\tilde{\mathbf{w}}^{0,t} = \mathbf{w}^{0,t} = \mathbf{v}^{0,t} = \tilde{\mathbf{w}}^{K,t-1}$ . In addition, let  $\tilde{\mathbf{w}}^{k,t} = \sum_n \frac{\sum_m \mathbb{1}_m^t a_{m,n} \tilde{D}_n}{\sum_n \sum_m \mathbb{1}_m^t a_{m,n} \tilde{D}_n} \tilde{\mathbf{w}}_n^{k,t}$  and  $\mathbf{w}^{k,t} = \sum_{n \in \mathcal{N}} \frac{D_n}{\sum_{n \in \mathcal{N}} D_n} \mathbf{w}_n^{k,t}$  denote the weighted average of the sets of model parameters  $\tilde{\mathbf{w}}_n^{k,t}$  and  $\mathbf{w}_n^{k,t}$ , respectively.

To facilitate the analysis, we make the following assumptions on the loss function to describe how the data is distributed at different devices.

**Assumption 1**: For each data point  $\{\mathbf{x}_i, y_i\} \in \mathcal{D}_n$ , the gradient of the function  $f(\mathbf{w}, \{\mathbf{x}_i, y_i\})$  has bounded variance, i.e.,  $\mathbb{E} \|\nabla f(\mathbf{w}, \{\mathbf{x}_i, y_i\}) - \nabla f(\mathbf{w}, \mathcal{D}_n)\| \leq \sigma_n$ .

**Assumption 2**: For each device, the gradient of the local loss function  $f(\mathbf{w}, \mathcal{D}_n)$  and the global loss function  $f(\mathbf{w}, \cup \mathcal{D}_n)$  satisfy  $\|\nabla f(\mathbf{w}, \mathcal{D}_n) - \nabla f(\mathbf{w}, \cup \mathcal{D}_n)\| \leq \delta_n$ .

Based on **Assumption 1** and **2**, we investigate model divergence  $\|\hat{\mathbf{w}}_m^t - \mathbf{v}^{K,t}\|$  in **Theorem 1**.

**Theorem 1**: Assume that the local loss function  $f(\mathbf{w}, \mathcal{D}_n)$  is  $L_n$ -smooth. The divergence between  $\hat{\mathbf{w}}_m^t$  and  $\mathbf{v}^{K,t}$  in the  $t$ -th communication round can be written as

$$\|\hat{\mathbf{w}}_m^t - \mathbf{v}^{K,t}\| \leq \Phi_m \triangleq \sum_{n \in \mathcal{N}} \frac{a_{m,n} \tilde{D}_n}{\sum_{n \in \mathcal{N}} a_{m,n} \tilde{D}_n} \left( \frac{\sigma_n}{L_n \sqrt{\tilde{D}_n}} \right)$$

$$+ \frac{\delta_n}{L_n} \left) ((\beta L_n + 1)^k - 1). \quad (12)$$

*Proof:* Please see Appendix A.  $\blacksquare$

**Theorem 1** reveals the impact of data distribution on FL performance, wherein lower variances  $\sigma_n$  and  $\delta_n$  produce better training performance. That is, the gateway and associated devices are more helpful for FL training if the local data distribution better represents the overall data distribution. In addition, we find that larger training data size  $\tilde{D}_n$  can lead to smaller divergence. Moreover, the divergence increases with the value of local epoch  $K$ , which follows the same trend with the standard FL framework [32].

According to the model divergence bound in (12), we derive the proportion of the  $m$ -th gateway and its associated devices' participation rate over the total participation rate as  $\frac{1/\Phi_m}{\sum_{m \in \mathcal{M}} 1/\Phi_m}$ . Recall that we select  $J$  gateways and the associated devices to participate in the local model training in each communication round. In this context, the total participation rate of all gateways and the associated devices is  $J$ . As such, the participation rate of the  $m$ -th gateway and its associated devices is determined by [33]–[35]

$$\Gamma_m = \min \left\{ J \frac{1/\Phi_m}{\sum_{m \in \mathcal{M}} 1/\Phi_m}, 1 \right\}. \quad (13)$$

Note that the participation rate of each gateway and its associated devices cannot exceed 1.

This participation rate  $\Gamma_m$  derived by the divergence bound of the  $m$ -th gateway and associated devices is introduced to show how many communication rounds that the  $m$ -th gateway should participate in the whole FL process. Based on the derived participation rate  $\Gamma_m$ , we can optimize the communication and energy resources while guaranteeing FL training performance by adopting a device-specific participation rate constraint. Superior to the general fairness guarantee (e.g. Round Robin), the participation rate constraint can not only save the slow devices from being excluded from FL training process, but also involve important devices with better data distribution in more communication rounds on the track of low latency by setting a larger participation rate for the important devices.

## V. DYNAMIC DEVICE SCHEDULING AND RESOURCE ALLOCATION ALGORITHM

In this section, we propose a dynamic device scheduling and resource allocation (DDsRA) algorithm to solve the stochastic optimization problem **P0**, which is shown in **Algorithm 1**. The proposed DDsRA as a centralized scheduling algorithm is performed by the BS. Compared with the existing DNN partition approaches using a predefined DNN partition point for all devices during the FL training process [19]–[21], the proposed DDsRA algorithm dynamically optimizes DNN partition point, channel assignment, transmit power, and computation frequency with time-varying channels and stochastic energy arrivals.

---

### Algorithm 1: Dynamic device scheduling and resource allocation algorithm

---

- 1 Initialize: Virtual queue length  $\mathbf{Q}(t) = 0$ ;
  - 2 **for**  $t = 1, 2, \dots, T$  **do**
  - 3     **Require:** Virtual queue length and channel state at the beginning of the  $t$ -th communication round;
  - 4     **Ensure:**  $\mathbf{X}(t) = [\mathbf{I}(t), \mathbf{l}(t), \mathbf{P}(t), \mathbf{f}^G(t)]$ ;
  - 5     **do in parallel**
  - 6         Optimize DNN partition point  $\mathbf{l}(t)$ , computation frequency  $\mathbf{f}^G(t)$  and transmit power  $\mathbf{P}(t)$  by solving (21), (22), and (23) with block coordinate descent method, and compute  $\Lambda_{m,j}(t)$  according to (18);
  - 7     Given the optimized auxiliary variable  $\Lambda_{m,j}(t)$ , find the channel assignment policy  $\mathbf{I}(t)$  by solving (26) with Hungarian method;
  - 8     Update  $\mathbf{Q}(t)$  according to (14);
  - 9     **Return**  $\mathbf{X}(t) = [\mathbf{I}(t), \mathbf{l}(t), \mathbf{P}(t), \mathbf{f}^G(t)]$
- 

### A. Problem Transformation

Based on the Lyapunov optimization method [36], we first transform the original problem **P0** into **P1** by converting the time-average inequality constraint **C11** to the queue stability constraint **C11'**. To this end, we define the virtual queue  $Q_m(t)$  for each gateway updated by

$$Q_m(t+1) \triangleq \max \{Q_m(t) - \mathbb{1}_m^t + \Gamma_m, 0\}. \quad (14)$$

By replacing the long-term participation rate constraint **C11** with mean rate stability constraint of  $Q_m(t)$ , the original problem **P0** can be written as

$$\begin{aligned} \mathbf{P1} : \max_{\mathbf{X}(t)} & \frac{1}{T} \sum_{t=1}^T \tau(t) \\ \text{s.t. } & \mathbf{C1} \sim \mathbf{C10}, \mathbf{C11}' : \lim_{t \rightarrow \infty} \frac{\mathbb{E}\{|Q_m(t)|\}}{t} = 0, \forall m \in \mathcal{M}. \end{aligned} \quad (15)$$

To solve **P1**, we next transform the long-term stochastic problem **P1** into the static problem **P2** in each communication round by means of characterizing the Lyapunov drift-plus-penalty function [36].

**Definition 1:** Given  $V > 0$ , the Lyapunov drift-plus-penalty function is defined as

$$\Delta_V(t) \triangleq V\tau(t) + \Delta\Xi(t), \quad (16)$$

where  $\Delta\Xi(t) \triangleq \mathbb{E}\{\Xi(t+1) - \Xi(t) | \mathbf{Q}(t)\}$  is the conditional Lyapunov drift, and  $\Xi(t) \triangleq \frac{1}{2} \sum_{m \in \mathcal{M}} Q_m(t)^2$  is the Lyapunov function.

Minimizing  $\Delta\Xi(t)$  stabilizes the virtual queues  $\mathbf{Q}(t)$  and encourages the virtual queues to meet the mean rate stability constraint **C11'** [37]. As such, minimizing the Lyapunov drift-plus-penalty function can concurrently minimize the FL delay and satisfy the long-term participation rate constraint **C11**, where  $V$  is a control parameter to tune the trade-off between latency minimization and the degree of which the long-term participation rate constraint is satisfied.

**Lemma 1:** Given the virtual queue lengths  $\mathbf{Q}(t)$ ,  $\Delta\Xi(t)$  is upper bounded by  $\Delta\Xi(t) \leq H + \sum_{m \in \mathcal{M}} \mathbb{E}\{Q_m(t)(\Gamma_m - \mathbb{1}_m^t) | \mathbf{Q}(t)\}$ , where  $H = \frac{1}{2} \sum_{m \in \mathcal{M}} (\Gamma_m + 1)$ .

*Proof:* Please see Appendix B. ■

Thus, the DDSRA algorithm is proposed to minimize the Lyapunov drift-plus-penalty function  $\Delta_V(t)$  in (16) in each communication round, i.e.,

$$\mathbf{P2}: \min_{\mathbf{X}(t)} V\tau(t) - \sum_{m \in \mathcal{M}} \sum_{j \in \mathcal{J}} Q_m(t) I_{m,j}(t) \quad (17)$$

s.t. **C1** ~ **C10**.

### B. Optimal Solution of P2

To solve **P2**, we first introduce an  $M \times J$  matrix  $\Lambda(t)$  of auxiliary variables

$$\Lambda_{m,j}(t) = \max_{n \in \mathcal{N}_m} \left\{ \left( \frac{\sum_{l=1}^{l_n(t)} (o_l + o'_l)}{\phi_n^D f_n^D} + \frac{\sum_{l=l_n(t)+1}^L (o_l + o'_l)}{\phi_m^G f_{m,n}^G(t)} \right) K \tilde{D}_n \right\} + \gamma / B^d / \log_2 \left( 1 + \frac{P^B h_{m,j}^d(t)}{B^d N_0 + i_{m,j}^d(t)} \right) + \gamma / B^u / \log_2 \left( 1 + \frac{P_m(t) h_{m,j}^u(t)}{B N_0 + i_{m,j}^u(t)} \right). \quad (18)$$

Note that  $\Lambda_{m,j}(t)$  represents the total delay for the  $m$ -th gateway if it is assigned to the  $j$ -th channel in the  $t$ -th communication round, and  $\mathcal{N}_m \subset \mathcal{N}$  denotes the index set of the devices associated with the  $m$ -th gateway. As such, **P2** can be rewritten as

$$\mathbf{P3}: \min_{\mathbf{X}(t)} V \max_{m \in \mathcal{M}} \left\{ \sum_{j \in \mathcal{J}} I_{m,j}(t) \Lambda_{m,j}(t) \right\} - \sum_{m \in \mathcal{M}} \sum_{j \in \mathcal{J}} Q_m(t) I_{m,j}(t) \quad (19)$$

s.t. **C1** ~ **C10**.

By exploiting the independence between  $\mathbf{I}(t)$  and  $\Lambda(t)$  in the objective function of **P3**, we decouple the joint optimization problem into the following sub-problems.

1) *Optimal auxiliary variable:* Since  $\Lambda(t)$  is independent of  $\mathbf{I}(t)$  in **P3**, we can separately minimize  $\Lambda_{m,j}(t)$  by optimizing the corresponding DNN partition point  $l_n(t)$ , transmitting power  $P_m(t)$ , and computation frequency  $f_{m,n}^G(t)$  as

$$\min_{l_n(t), P_m(t), f_{m,n}^G(t), \forall n \in \mathcal{N}_m} \Lambda_{m,j}(t) \quad (20)$$

s.t. **C4** ~ **C6**,

$$\mathbf{C7}': \sum_{l=1}^{l_n(t)} g_{n,l} \leq G_n^{\text{D,max}}, \forall n \in \mathcal{N}_m,$$

$$\mathbf{C8}': \sum_{n \in \mathcal{N}_m} \sum_{l=l_n(t)+1}^L g_{n,l} \leq G_m^{\text{G,max}},$$

$$\mathbf{C9}': \sum_{n \in \mathcal{N}_m} K \tilde{D}_n \frac{v_m^G}{\phi_m^G} \left( \sum_{l=l_n(t)+1}^L (o_l + o'_l) \right) (f_{m,n}^G(t))^2 + \frac{\gamma P_m(t)}{B^u \log_2 \left( 1 + \frac{P_m(t) B h_{m,j}^u(t)}{B^u N_0 + i_{m,j}^u(t)} \right)} \leq E_m^G(t),$$

$$\mathbf{C10}': K \tilde{D}_n \frac{v_n^D}{\phi_n^D} \left( \sum_{l=1}^{l_n(t)} (o_l + o'_l) \right) (f_n^D)^2 \leq E_n^D(t), \forall n \in \mathcal{N}_m.$$

To solve this problem, we decompose (20) into three sub-problems in (21), (22) and (23). Given the remaining variables, each sub-problem is solved by the bisection method or successive convex optimization method [38]. Thus, (20) can be optimized by the block coordinate descent method as shown in **Algorithm 1**.

Given the optimized  $P_m(t)$  and  $f_{m,n}^G(t)$ , we can rewrite (20) as

$$\min_{l_n(t), \forall n \in \mathcal{N}_m} g_1(l_n(t)) = \max_{n \in \mathcal{N}_m} \left\{ K \tilde{D}_n \left( \frac{\sum_{l=1}^{l_n(t)} (o_l + o'_l)}{\phi_n^D f_n^D} + \frac{\sum_{l=l_n(t)+1}^L (o_l + o'_l)}{\phi_m^G f_{m,n}^G(t)} \right) \right\} \quad (21)$$

s.t. **C5**, **C7'**, **C8'**, **C9'**, **C10'**.

Note that the sub-problem in (21) is NP-hard. To circumvent this difficulty, a greedy solution with polynomial-time complexity is proposed by adopting the bisection method [39]. Let  $g_1^{\min} = \frac{K \min_{n \in \mathcal{N}_m} \{\tilde{D}_n\} \sum_{l=1}^{l_n(t)} (o_l + o'_l)}{\max\{\max_{n \in \mathcal{N}_m} \{\phi_n^D f_n^D\}, \max_{n \in \mathcal{N}_m} \{\phi_m^G f_{m,n}^G(t)\}\}}$  and  $g_1^{\max} = \frac{K \max_{n \in \mathcal{N}_m} \{\tilde{D}_n\} \sum_{l=1}^{l_n(t)} (o_l + o'_l)}{\min\{\min_{n \in \mathcal{N}_m} \{\phi_n^D f_n^D\}, \min_{n \in \mathcal{N}_m} \{\phi_m^G f_{m,n}^G(t)\}\}}$  denote the lower and upper bound of  $g_1(l_n(t))$ . Let  $\eta$  be the mid point of the interval  $(g_1^{\min}, g_1^{\max})$ , i.e.,  $\eta = \frac{1}{2}(g_1^{\min} + g_1^{\max})$ . In each iteration, we first compute the lower and upper bound of  $l_n(t)$  according to constraints **C5**, **C7'**, **C9'** and  $K \tilde{D}_n \left( \frac{\sum_{l=1}^{l_n(t)} (o_l + o'_l)}{\phi_n^D f_n^D} + \frac{\sum_{l=l_n(t)+1}^L (o_l + o'_l)}{\phi_m^G f_{m,n}^G(t)} \right) \leq \eta, \forall n \in \mathcal{N}_m$ , i.e.,  $l_n^{\min} \leq l_n(t) \leq l_n^{\max}$ . If constraints **C8'** and **C10'** hold when  $l_n(t) = l_n^{\min}$ , we refine the upper bound of  $g_1(l_n(t))$  as  $\eta$ . Otherwise, the lower bound of  $g_1(l_n(t))$  is refined as  $\eta$ . We can note that  $l_n^{\min} = l_n^{\max}$  if the bisection method converges. Thus, the optimal DNN partition point is derived as  $l_n^*(t) = l_n^{\min}$ .

Given the optimized  $l_n(t)$  and  $P_m(t)$ , we can rewrite (20) as

$$\min_{f_{m,n}^G(t), \forall n \in \mathcal{N}_m} g_2(f_{m,n}^G(t)) = \max_{n \in \mathcal{N}_m} \left\{ K \tilde{D}_n \left( \frac{\sum_{l=1}^{l_n(t)} (o_l + o'_l)}{\phi_n^D f_n^D} + \frac{\sum_{l=l_n(t)+1}^L (o_l + o'_l)}{\phi_m^G f_{m,n}^G(t)} \right) \right\} \quad (22)$$

s.t. **C6**, **C10'**.

Similarly, the sub-problem in (22) can be solved by the bisection method. Let  $g_2^{\min} = K \min_{n \in \mathcal{N}_m} \{\tilde{D}_n\} \left( \frac{\sum_{l=1}^{l_n(t)} (o_l + o'_l)}{\max_{n \in \mathcal{N}_m} \{\phi_n^D f_n^D\}} + \frac{\sum_{l=l_n(t)+1}^L (o_l + o'_l)}{\phi_m^G f_{m,n}^{\text{G,max}}(t)} \right)$  and  $g_2^{\max} = K \max_{n \in \mathcal{N}_m} \{\tilde{D}_n\} \left( \frac{\sum_{l=1}^{l_n(t)} (o_l + o'_l)}{\min_{n \in \mathcal{N}_m} \{\phi_n^D f_n^D\}} + \frac{\sum_{l=l_n(t)+1}^L (o_l + o'_l)}{\phi_m^G f_{m,n}^{\text{G,min}}(t)} \right)$  denote the lower and upper bound of  $g_2(f_{m,n}^G(t))$ . Let  $\vartheta$  be the mid point of the interval  $(g_2^{\min}, g_2^{\max})$ , i.e.,  $\vartheta = \frac{1}{2}(g_2^{\min} + g_2^{\max})$ . In each iteration, we first compute the lower bound of  $f_{m,n}^G(t)$  according to  $K \tilde{D}_n \left( \frac{\sum_{l=1}^{l_n(t)} (o_l + o'_l)}{\phi_n^D f_n^D} + \frac{\sum_{l=l_n(t)+1}^L (o_l + o'_l)}{\phi_m^G f_{m,n}^G(t)} \right) \leq \vartheta$ , i.e.,  $f_{m,n}^G(t) \geq \left( \sum_{l=l_n(t)+1}^L (o_l + o'_l) \right) / \phi_m^G / \left( - \frac{\sum_{l=1}^{l_n(t)} (o_l + o'_l)}{\phi_n^D f_n^D} + \frac{\vartheta}{K \tilde{D}_n} \right)$ . If constraints **C6** and **C10'** hold when  $f_{m,n}^G(t) = \left( \sum_{l=l_n(t)+1}^L (o_l + o'_l) \right) / \phi_m^G / \left( - \frac{\sum_{l=1}^{l_n(t)} (o_l + o'_l)}{\phi_n^D f_n^D} + \frac{\vartheta}{K \tilde{D}_n} \right)$ , we refine the upper bound of  $g_2(f_{m,n}^G(t))$  as  $\vartheta$ . Otherwise, the

lower bound of  $g_2(f_{m,n}^G(t))$  is refined as  $\vartheta$ . Suppose that  $\vartheta = \vartheta^*$  when the bisection method converges. Thus, the optimal computation frequency is derived as  $f_{m,n}^{G^*} = (\sum_{l=l_n(t)+1}^L (o_l + o'_l)) / \phi_m^G / \left( \frac{\vartheta^*}{K\tilde{D}_n} - \frac{\sum_{l=l_n(t)+1}^L (o_l + o'_l)}{\phi_n^D f_n^D} \right)$ .

Given the optimized  $l_n(t)$  and  $f_{m,n}^G(t)$ , we rewrite (20) as

$$\min_{P_m(t), \forall n \in \mathcal{N}_m} g_3(P_m(t)) = \frac{\gamma}{B^u \log_2 \left( 1 + \frac{P_m(t) h_{m,j}^u(t)}{B^u N_0 + i_{m,j}^u(t)} \right)} \quad (23)$$

s.t. **C4**, **C10'**.

Note that the sub-problem in (23) is convex. The optimal transmit power is as follows.

$$P_m^*(t) = \begin{cases} 0, & \text{if } \frac{B^u}{\gamma \ln 2} \left( E_m^G(t) - \sum_{n \in \mathcal{N}_m} K \tilde{D}_n \frac{v_m^G}{\phi_m^G} \left( \sum_{l=l_n(t)+1}^L (o_l + o'_l) \right) (f_{m,n}^G(t))^2 \right) - \frac{B N_0 + i_{m,j}^u(t)}{h_{m,j}^u(t)} \leq 0, \\ \min\{x^*, P_m^{\max}\}, & \text{otherwise,} \end{cases} \quad (24)$$

where  $x^* > 0$  is the solution to  $\frac{B^u}{\gamma} \left( E_m^G(t) - \sum_{n \in \mathcal{N}_m} K \tilde{D}_n \frac{v_m^G}{\phi_m^G} \left( \sum_{l=l_n(t)+1}^L (o_l + o'_l) \right) (f_{m,n}^G(t))^2 \right) \log_2 \left( 1 + \frac{h_{m,j}^u(t)x}{B^u N_0 + i_{m,j}^u(t)} \right) - x = 0$ .

2) *Optimal channel assignment*: Given the optimized  $\Lambda(t)$ , the channel assignment matrix  $\mathbf{I}(t)$  can be optimized as

$$\min_{\mathbf{I}(t)} V \max_{m \in \mathcal{M}} \left\{ \sum_{j \in \mathcal{J}} I_{m,j}(t) \Lambda_{m,j}(t) \right\} - \sum_{m \in \mathcal{M}} \sum_{j \in \mathcal{J}} Q_m(t) I_{m,j}(t) \quad (25)$$

s.t. **C1** ~ **C3**.

To solve the problem in (25), we first introduce an auxiliary variable  $\lambda$ , and thus the problem can be equivalently transformed into

$$\min_{\lambda, \mathbf{I}(t)} \lambda - \sum_{m \in \mathcal{M}} \sum_{j \in \mathcal{J}} Q_m(t) I_{m,j}(t) \quad (26)$$

s.t. **C1** ~ **C3**, **C12**:  $\lambda \geq V \sum_{j \in \mathcal{J}} I_{m,j}(t) \Lambda_{m,j}(t), \forall m \in \mathcal{M}$ .

Following the solution of the problem in (20), we decompose (26) into two sub-problems in (27) and (30), and then optimize the auxiliary variable  $\lambda$  and the channel assignment matrix  $\mathbf{I}(t)$  by solving the sub-problems in an iterative manner.

Given the optimized auxiliary variable  $\lambda$ , we can rewrite (26) as

$$\min_{\mathbf{I}(t)} - \sum_{m \in \mathcal{M}} \sum_{j \in \mathcal{J}} Q_m(t) I_{m,j}(t) \quad (27)$$

s.t. **C1**:  $I_{m,j}(t) \in \{0, 1\}, \forall m \in \mathcal{M}, j \in \mathcal{J}$ ,

**C2**:  $\sum_{j \in \mathcal{J}} I_{m,j}(t) \leq 1, \forall m \in \mathcal{M}$ ,

**C3**:  $\sum_{m \in \mathcal{M}} I_{m,j}(t) = 1, \forall j \in \mathcal{J}$ ,

**C12**:  $\sum_{j \in \mathcal{J}} V \Lambda_{m,j}(t) I_{m,j}(t) \leq \lambda, \forall m \in \mathcal{M}$ .

From constraints **C1** and **C2**, constraint **C12** can be equivalently transformed into **C12'**, i.e.,  $I_{m,j}(t) = 0, \forall (m, j) \in \{(m, j) \in \mathcal{M} \times \mathcal{J} | \Lambda_{m,j}(t) > \lambda/V\}$ . By replacing the corresponding weights  $Q_m(t)$  in the objective function of (27) with an extremely large positive value, the problem in (27) can be transformed into a standard weighted bipartite matching linear program as

$$\min_{\mathbf{I}(t)} \sum_{m \in \mathcal{M}} \sum_{j \in \mathcal{J}} \Theta_{m,j} I_{m,j}(t) \quad (28)$$

s.t. **C1**:  $I_{m,j}(t) \in \{0, 1\}, \forall m \in \mathcal{M}, j \in \mathcal{J}$ ,

**C2**:  $\sum_{j \in \mathcal{J}} I_{m,j}(t) \leq 1, \forall m \in \mathcal{M}$ ,

**C3**:  $\sum_{m \in \mathcal{M}} I_{m,j}(t) = 1, \forall j \in \mathcal{J}$ ,

where

$$\Theta_{m,j} = \begin{cases} \Psi, & \text{if } (m, j) \in \{(m, j) \in \mathcal{M} \times \mathcal{J} | V \Lambda_{m,j}(t) > \lambda\}, \\ -Q_m(t), & \text{otherwise.} \end{cases} \quad (29)$$

Note that  $\Psi$  is set as an extremely large positive value to create the composite objective function in (28) which incorporates the effect of constraint **C12'**. Based on the Hungarian method [40], [41], the optimal channel assignment matrix  $\mathbf{I}^*(t)$  can be obtained in polynomial time.

Given the optimized channel assignment matrix  $\mathbf{I}(t)$ , we can rewrite (26) as

$$\min_{\lambda} \lambda \quad (30)$$

s.t. **C12**:  $\lambda \geq V \sum_{j \in \mathcal{J}} I_{m,j}(t) \Lambda_{m,j}(t), \forall m \in \mathcal{M}$ .

Obviously, the optimal auxiliary variable is given by

$$\lambda^* = \max_{m \in \mathcal{M}} \left\{ \sum_{j \in \mathcal{J}} I_{m,j}(t) \Lambda_{m,j}(t) \right\}. \quad (31)$$

With the optimal channel assignment matrix  $\mathbf{I}^*(t)$ , whether the  $m$ -th gateway and its associated devices are selected to participate in the local model training in the  $t$ -th communication round is determined by  $\mathbb{1}_m^t = \sum_{j \in \mathcal{J}} i_{m,j}^*(t)$ .

### C. Optimality, Complexity, Applicability, and Scalability Analysis

In this subsection, we present the optimality, complexity, applicability, and scalability analysis of the proposed DDSRA algorithm as follows.

*Optimality analysis*: The DDSRA algorithm converges to at least a locally optimal solution. From Section V-B, the DDSRA algorithm consists of two parts: a) solve the auxiliary variables  $\Lambda_{m,j}(t)$  in (20) based on block coordinate descent method in the outer layer loop and bisection method in the inner layer loop, and b) solve the channel assignment matrix  $\mathbf{I}(t)$  in (26) based on block coordinate descent method in the outer layer loop and Hungarian method in the inner layer loop. According to the existing works on the block coordinate descent method [42]–[44], the convergence to a local optimum can be guaranteed when the sub-problems in each iteration can

be solved exactly with optimality. Notably, bisection method is an efficient and widely-used algorithm which can converge to the global optimum superlinearly [45], [46], and Hungarian method is a straightforward method of finding the optimal solution to an assignment problem [40], [41]. According to the above analysis, the DDSRA algorithm converges to at least a locally optimal solution.

*Complexity analysis:* Let  $L_1$  and  $L_2$  denote the required number of iterations for solving (20) based on the block coordinate descent method in the outer layer loop and the bisection method in the inner layer loop, respectively. The computational complexity of solving the auxiliary variables  $\Lambda_{m,j}(t)$  is represented as  $\mathcal{O}(NJJL_1L_2)$ . From Section V-B, the computational complexity of the Hungarian method in the inner layer is represented as  $\mathcal{O}(M^3)$ . Let  $L_3$  denote the required number of iterations for solving (26) based on the block coordinate descent method in the outer layer loop. The computational complexity of solving the channel assignment matrix  $\mathbf{I}(t)$  in (26) is represented as  $\mathcal{O}(M^3L_3)$ . To sum up, the total computational complexity of DDSRA is  $\mathcal{O}(NJJL_1L_2 + M^3L_3)$ .

*Applicability analysis:* The DDSRA algorithm is applicable to a variety of IIoT scenarios. Thanks to the joint optimization of device scheduling and resource allocation (i.e. DNN partition point, channel assignment, transmit power, and computation frequency), our proposal can be potentially applied in device heterogeneity scenarios wherein IIoT devices are intrinsically heterogeneous in computational capacity and memory resource. Thanks to the developed device-specific participation rate linked to the training dataset size and data distribution, our proposal is robust against data heterogeneity (i.e., non-IID data distribution). Moreover, thanks to the layer-level memory usage and FLOPs calculation model, our proposal is applicable to diverse DNN models such as multilayer perceptron (MLP) and convolutional neural network (CNN).

*Scalability analysis:* The computational complexity of the DDSRA algorithm is directly proportional to the number of end devices, i.e.,  $N$ . Furthermore, by exploiting the independence between the auxiliary variables  $\Lambda_{m,j}(t)$ , the DDSRA algorithm is decomposed into  $MJ$  multi-threaded parallel computation tasks (see line 5 in **Algorithm 1**), which can greatly reduce the time complexity. Therefore, the DDSRA algorithm is scalable to a large number of end devices.

## VI. PERFORMANCE ANALYSIS

### A. Asymptotic Optimality of DDSRA

In this subsection, we will analyze the performance of the proposed DDSRA algorithm in terms of asymptotic optimality, and characterize the trade-off between the delay minimization and the degree of which the participation rate constraint is satisfied.

**Theorem 2:** With the optimal policy of  $\mathbf{P2}$  in each communication round, and note that  $\mathbb{E}\{Q(0)\} < \infty$ , we have

$$\varphi^* - \varphi^{\text{opt}} \leq \frac{H}{V} + \frac{\mathbb{E}\{\Xi(0) - \Xi(T)\}}{VT}, \quad (32)$$

and

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{1}_m^t \geq \Gamma_m - \sqrt{\frac{H+V(\varphi^{\text{opt}} - \tau^{\text{min}})}{T} + \sum_{m \in \mathcal{M}} \frac{\mathbb{E}\{Q_m(0)^2\}}{T^2}}, \quad (33)$$

where  $\varphi^{\text{opt}}$  is the optimal utility of  $\mathbf{P0}$  over all possible scheduling policies,  $\varphi^*$  represents the optimal utility of  $\mathbf{P2}$ , and  $\tau^{\text{min}} = \frac{K \min_{n \in \mathcal{N}} \{\bar{D}_n\} \sum_{l=1}^L (o_l + o'_l)}{\min\{\min_{n \in \mathcal{N}} \{\phi_n^d f_n^d\}, \min_{m \in \mathcal{M}} \{\phi_m^g f_m^{\text{G,max}}\}\}} + \gamma/B^u / \log_2(1 + P_m^{\text{max}} h_{m,j}^u / (B^u N_0 + i_{m,j}^u)) + \gamma/B^d / \log_2(1 + P^B h_{m,j}^d / (B^d N_0 + i_{m,j}^d))$ .

*Proof:* Please see Appendix C. ■

We have verified the asymptotic optimality of the proposed DDSRA algorithm in (32). That is, the proposed DDSRA algorithm converges to the optimal solution as  $V$  increases. Moreover, (33) indicates that the participation rate of each gateway and its associated devices increases, and finally converges to the optimized device-specific participation rate  $\Gamma_m$  as  $V$  decreases. Hence, **Theorem 2** shows an  $[\mathcal{O}(1/V), \mathcal{O}(\sqrt{V})]$  trade-off between the minimization of FL training latency and the degree of which the participation rate constraint is satisfied, where the control parameter  $V$  represents how much we emphasize the maximization of the FL training latency. To be specific, a large value of  $V$  encourages reducing the FL training latency, which can be adopted for real-time delay-sensitive IIoT applications. Meanwhile, a small value of  $V$  pushes the participation rate of each gateway and its associated devices to the optimized device-specific participation rate  $\Gamma_m$ , thereby promoting the FL training performance.

### B. FL Convergence Analysis of the DDSRA Algorithm

For ease of exposition, we define  $\delta = \max_n \{\delta_n\}$ ,  $\sigma = \max_n \{\sigma_n\}$ ,  $F_n(\mathbf{w}) = f(\mathbf{w}, \mathcal{D}_n)$ ,  $\tilde{F}_n(\mathbf{w}) = f(\mathbf{w}, \tilde{\mathcal{D}}_n)$ , and  $F(\mathbf{w}) = f(\mathbf{w}, \cup \mathcal{D}_n)$ . Based on **Assumption 1** and **2**, the FL convergence bound of the proposed algorithm is derived as follows.

**Theorem 3:** Assume that the loss function  $F_n(w)$  is convex,  $L_n$ -smooth and  $\rho_n$ -Lipschitz continuous, the FL convergence bound is represented as

$$\mathbb{E}[F(\mathbf{W}^T) - F(\mathbf{w}^*)] \leq \frac{1}{T \left( \beta \phi - \frac{\rho \left( \delta + \sum_{n \in \mathcal{N}} \xi_n \frac{\sigma_n}{\sqrt{\bar{D}_n}} \right) ((\beta L + 1)^K - 1) + \beta \left( \delta + \sum_{n \in \mathcal{N}} \left| \xi_n - \frac{\bar{D}_n}{\sum_{n \in \mathcal{N}} \bar{D}_n} \right| \rho_n \right)}{\varepsilon^2 KL} \right)}, \quad (34)$$

where  $L = \max_n \{L_n\}$ ,  $\rho = \max_n \{\rho_n\}$ ,  $\phi \triangleq \omega(1 - \beta L/2)$ ,  $\omega \triangleq \min_{t \in \mathcal{T}} \frac{1}{\|\mathbf{v}^{K,t-1} - \mathbf{w}^*\|^2}$ ,  $\xi_n = \frac{\sum_{m \in \mathcal{M}} \Gamma_m a_{m,n} \bar{D}_n}{\sum_{n \in \mathcal{N}} \sum_{m \in \mathcal{M}} \Gamma_m a_{m,n} \bar{D}_n}$ , and  $\varepsilon \triangleq \min_{t \in \mathcal{T}} [F(\mathbf{v}^{K,t}) - F(\mathbf{w}^*)]$ .

*Proof:* Please see Appendix D. ■

From **Theorem 3**, we can see that larger training data sizes  $\tilde{D}_n$  can reduce the value of the term  $\rho \left( \delta + \sum_n \xi_n \frac{\sigma_n}{\sqrt{\bar{D}_n}} \right) ((\beta L + 1)^K - 1)$  in (34), which contributes to a smaller convergence bound and thereby a better FL performance. In addition, by setting a larger participation rate for

important devices with better data distribution (i.e., lower variances  $\sigma_n$ ), our derived device-specific participation rate  $\Gamma_m$  can produce a lower weighted sum of  $\frac{\sigma_n}{\sqrt{D_n}}$ , thereby leading to a lower FL convergence bound. Moreover, it also shows that when the participation rate is set to be the same for each gateway and its associated devices, i.e.,  $\Gamma_m = \Gamma_{m'}, \forall m \neq m'$ , the term  $\left| \xi_n - \frac{D_n}{\sum_n D_n} \right|$  in (34) is zero if the training data sizes are proportionate to the local dataset sizes (i.e.,  $\tilde{D}_n = \alpha D_n, \forall n \in \mathcal{N}$ , where  $\alpha$  represents the data sampling ratio for the local datasets). That is, the training data sizes proportionate to the local dataset sizes can lead to a small convergence bound, thereby promoting a better FL performance.

In addition, we derive a convergence bound for non-convex setting in **Theorem 4**. **Theorem 4** below indicates that the proposed DDSRA algorithm can achieve a FL convergence rate of  $\mathcal{O}(1/T)$  for non-convex loss functions.

**Theorem 4:** Assume that the loss function  $F_n(w)$  is non-convex and  $L_n$ -smooth, the convergence bound is represented as

$$\begin{aligned} & \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} \left[ \|\nabla F(\tilde{w}^t)\|^2 \right] \leq \frac{2}{K\beta T} (\mathbb{E}[F(\tilde{w}^0)] - \mathbb{E}[F(\tilde{w}^T)]) + \\ & \frac{L\beta N}{T} \sum_{t=0}^{T-1} \sum_{n=1}^N \sum_{k=0}^{K-1} \left( \frac{\sum_{m \in \mathcal{M}} \Gamma_m a_{m,n} \tilde{D}_n}{\sum_{n \in \mathcal{N}} \sum_{m \in \mathcal{M}} \Gamma_m a_{m,n} \tilde{D}_n} \right)^2 \mathbb{E} \left[ \|\nabla F_n(\tilde{w}_n^{k,t})\|^2 \right] + \\ & \frac{N\beta^2}{KT} \sum_{t=0}^{T-1} \sum_{n=1}^N \sum_{k=0}^{K-1} \left( \frac{\sum_{m \in \mathcal{M}} \Gamma_m a_{m,n} \tilde{D}_n}{\sum_{n \in \mathcal{N}} \sum_{m \in \mathcal{M}} \Gamma_m a_{m,n} \tilde{D}_n} \right)^2 \\ & L_n^2 \beta^2 k \sum_{j=0}^{k-1} \mathbb{E} \left[ \|\nabla F_n(\tilde{w}_n^{j,t})\|^2 \right]. \end{aligned} \quad (35)$$

*Proof:* Please see Appendix E.  $\blacksquare$

## VII. EXPERIMENTAL RESULTS

### A. Experimental Setting

To evaluate the FL training performance of the proposed algorithm for complex datasets and DNNs, we utilize Street View House Numbers (SVHN) [47] and CIFAR-10 [48] datasets trained on VGG-11 [49] for non-IID setting to demonstrate the test accuracy performance.

- **SVHN.** SVHN contains over 600000  $32 \times 32$  RGB images in 10 classes (from 0 to 9), which is cropped from pictures of house number plates.
- **CIFAR-10.** The CIFAR-10 dataset consists of 60000  $32 \times 32$  RGB images in 10 classes (from 0 to 9), with 50000 training images and 10000 test images per class.

For non-IID setting, we follow the previous work [50] to distribute the data points in each local dataset. The data points are sorted by class and divided into two extreme cases: (a)  $q_m$ -class non-IID, where each device holds data points in  $q_m$  classes, and (b) IID, where each device holds data points in all of the 10 classes. In this experiment,  $q_m$  is randomly generated, and we set the non-IID degree of the data distribution (proportion of the  $q_m$ -class non-IID data points) as  $\chi = 1$ .

For comparison purpose, we also consider the following baseline schemes:

- **Random Scheduling** [26]. The BS uniformly selects  $J$  gateways and the associated devices at random for local model training in each communication round.
- **Round Robin** [26]. The BS divides the  $M$  gateways and the associated devices into  $\lceil \frac{J}{M} \rceil$  groups and consecutively assigns each group to the wireless channels in each communication round.
- **Loss Driven Scheduling.** The BS selects  $J$  gateways and the associated devices according to the local training loss for local model update in each communication round.
- **Delay Driven Scheduling.** The BS selects  $J$  gateways and the associated devices for local model training with the objective of minimizing FL latency in each communication round.

Besides, we consider  $M = 6$  gateways,  $N = 12$  devices, and  $J = 3$  channels. Each gateway is designed to be associated with 2 of the devices. For each device, the local dataset size  $D_n$  is uniformly distributed within  $(0, 2000]$ ,  $E_n^{\text{D,max}} = 5$  J,  $G_n^{\text{D,max}} = 2$  GB,  $f_n^{\text{D}}$  is uniformly distributed within  $[0.1, 1]$  GHz,  $\phi_n^{\text{D}} = 16$  FLOPs per CPU cycle [51], and  $v_n^{\text{D}} = 10^{-27}$ . For each gateway,  $d_m$  is uniformly distributed within  $[1000, 2000]$  m,  $E_m^{\text{G,max}} = 30$  J,  $G_m^{\text{G,max}} = 4$  GB,  $f_m^{\text{G,max}} = 4$  GHz,  $\phi_m^{\text{D}} = 32$  FLOPs per CPU cycle,  $v_m^{\text{G}} = 10^{-27}$ , and  $P_m^{\text{max}} = 200$  mW. The channel parameters are set as  $d_0 = 1$  m,  $\nu = 2$ ,  $B^{\text{u}} = 1$  MHz,  $B^{\text{d}} = 20$  MHz,  $N_0 = -174$  dBm/Hz,  $h_0 = -30$  dB,  $P^{\text{BS}} = 1$  W, the uplink and downlink interferences  $i_{m,j}^{\text{u}}(t)$  and  $i_{m,j}^{\text{d}}(t)$  are produced by the Gaussian distribution with different variances, and the channel power gains  $\rho_{m,n}^{\text{u}}(t)$  and  $\rho_{m,n}^{\text{d}}(t)$  are exponentially distributed with unit mean. For local model training, we set local epoch  $K = 5$ , training data sampling ratio  $\alpha = 0.05$ , and learning rate  $\beta = 0.01$ . The memory usage and FLOPs for the DNN layers trained at the device and gateway side can be calculated according to Table II. In addition, the values of  $L_n$ ,  $\sigma_n$ ,  $\delta_n$  and  $\rho_n$  are estimated by observing the model parameters in the FL training process.

### B. Performance of Device-specific Participate Rate Policy

To demonstrate the derived device-specific participation rate linked to FL performance in Section IV, we compare our derived participation rate of each gateway and the associated devices in (13) with the experimental value on SVHN and CIFAR-10 datasets, as shown in Fig.2. Note that the derived value is calculated based on the upper bound of the divergence between the local model parameters learned in the FL training process and the model parameters learned in the centralized training process, i.e.,  $\|\hat{w}_m^t - v^{K,t}\|$ , while the experimental value is obtained by observing  $\|\hat{w}_m^t - v^{K,t}\|$  in the training process. First, Fig.2 shows that the derived participation rate of each gateway and the associated devices is consistent with the experimental value, which justifies the divergence bound in **Theorem 1**. Second, Fig.2 shows that the 1-th gateway and the associated devices can achieve the highest participation rate. This is because we set each device associated with the

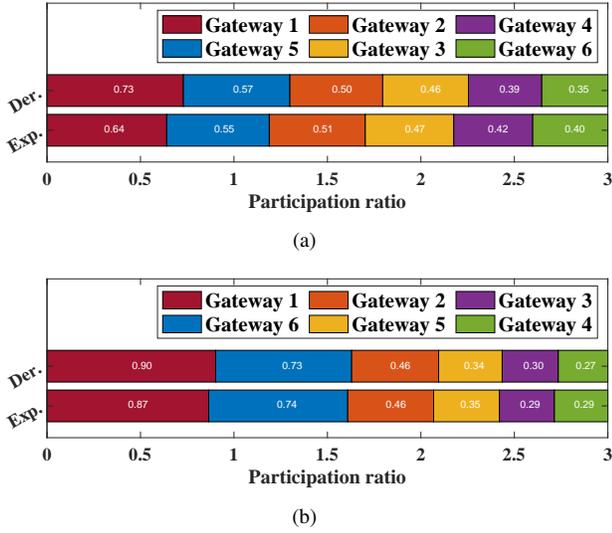


Fig. 2: The derived and experimental participation rate of each gateway and associated devices on (a) SVHN and (b) CIFAR-10 datasets.

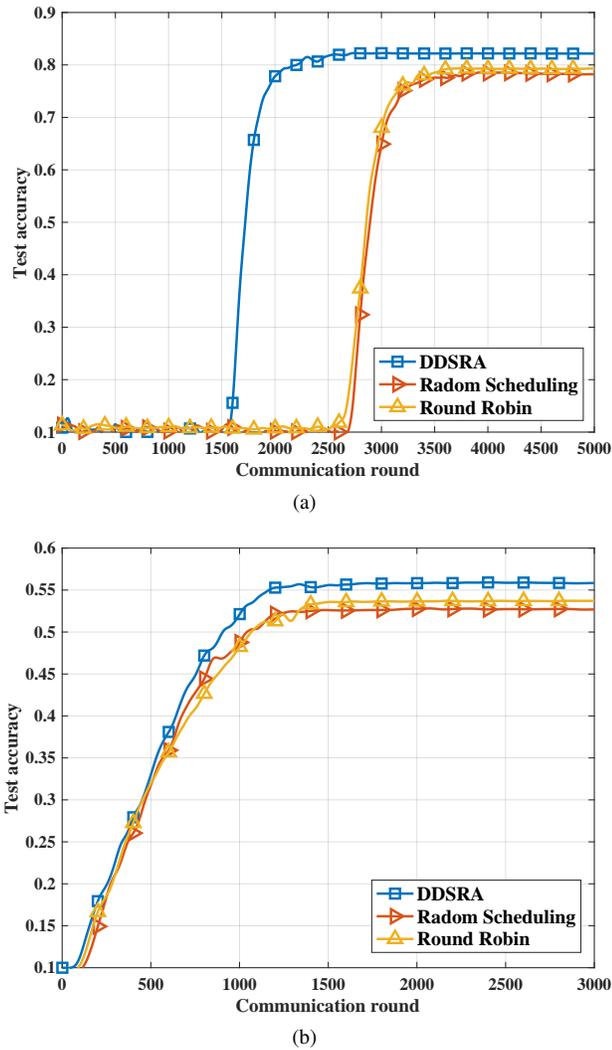


Fig. 3: Test accuracy comparison between the proposed device-specific participation rate policy and the baseline device scheduling policies (i.e., Random Scheduling policy and Round Robin policy) on (a) SVHN and (b) CIFAR-10 datasets.

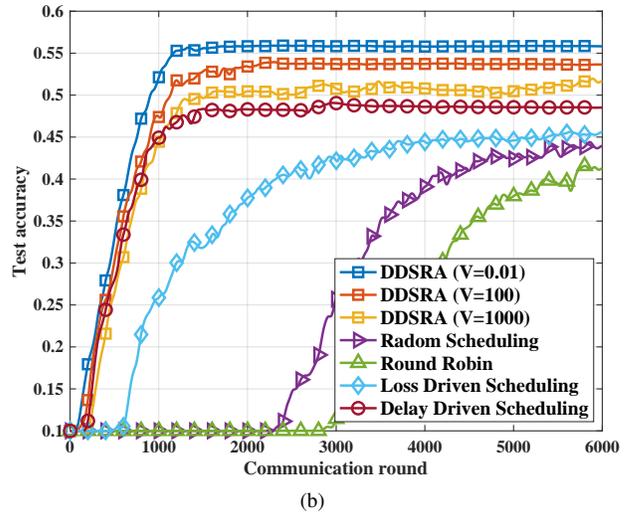
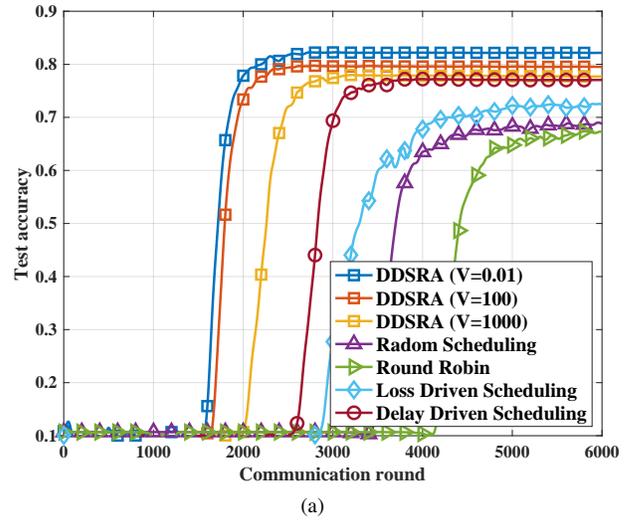


Fig. 4: Test accuracy comparison between DDSRA algorithm and the baseline schemes on (a) SVHN and (b) CIFAR-10 datasets.

1-th gateway a local dataset with a wider variety of the  $q_m$ -class non-IID data points, which makes the data distribution of the devices associated with the 1-th gateway better represents the overall data distribution. In addition, Fig.3 shows the comparison of the test accuracy between the proposed device-specific participation rate policy, Random Scheduling policy and Round Robin policy on SVHN and CIFAR-10 datasets. It can be observed that, with the same number of participant gateways and associated devices in each communication round, the proposed device-specific participation rate policy achieves better learning performance than the baseline schemes with fairness guarantee. Compared with Random Scheduling policy, the proposed device-specific participation rate policy reduces the number of communication rounds required for convergence by 35% for SVHN dataset, and improves the test accuracy by 6% for CIFAR-10 dataset.

### C. Performance of DDSRA Algorithm

Fig.4 and 5 show the test accuracy and the training delay comparison between the proposed DDSRA algorithm (with

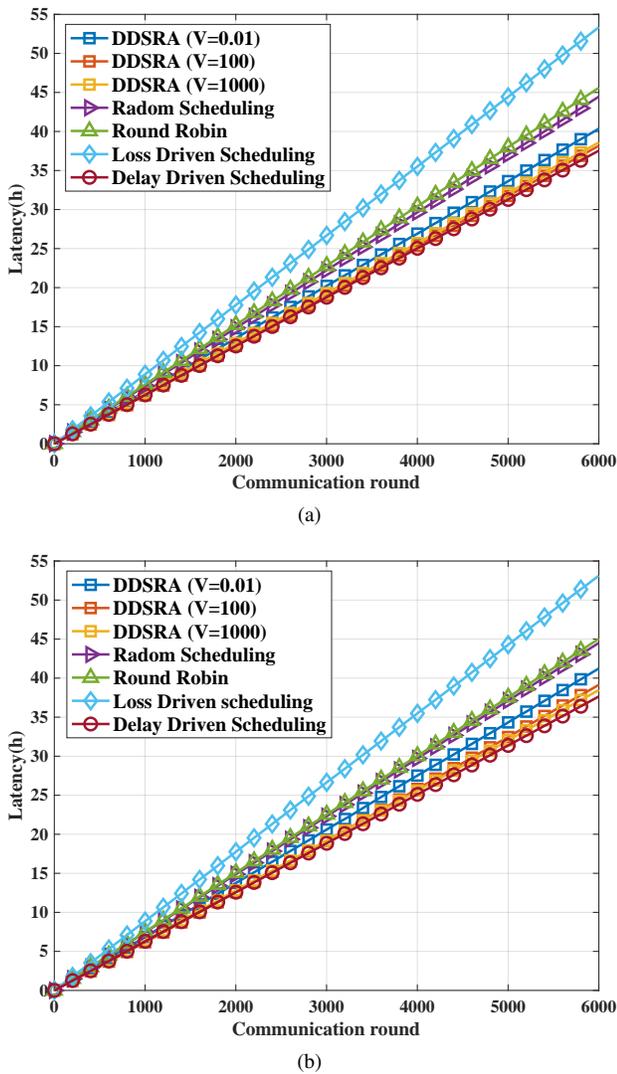


Fig. 5: Training delay comparison between DDSRA algorithm and the baseline schemes on (a) SVHN and (b) CIFAR-10 datasets.

$V = 0.01, 1000, \text{ and } 10000$ ) and the baseline schemes, i.e., Random Scheduling, Round robin, Loss Driven Scheduling, and Delay Driven Scheduling. First, we can observe that a smaller  $V$  can lead to a better FL performance but a higher FL training delay. It reveals that a smaller  $V$  guaranteeing the proper participation rate of each gateway and associated devices can obtain a higher test accuracy while prolonging the training delay, which conforms to **Theorem 2**. Second, it can be observed that, with limited energy supply and memory, the proposed algorithm can achieve an obvious advantage on test accuracy and convergence rate than baseline schemes. Compared with Round Robin, the DDSRA algorithm with  $V = 0.01$  reduces the number of communication rounds required for convergence by 53% for SVHN dataset and 78% for CIFAR-10 dataset, and improves the test accuracy by 22% for SVHN dataset and 37% for CIFAR-10 dataset, respectively. The intuition is that the proposed DDSRA algorithm guarantees a proper participation rate of each gateway and associated devices, which makes the local datasets with better data distribution more involved in the FL training process. In addition,

the joint communication, energy and memory resources allocation circumvents the local model training and transmitting failure due to the shortage of energy and memory, as such the gateways and devices can participate in more communication rounds to improve FL performance. Meanwhile, the baseline schemes fix the transmit power, computation frequency and the DNN partition point in the training process, as such devices and gateways often fail to complete the local model training and transmitting due to energy shortage. As a result, the low participation rate degrades the FL learning performance. Third, Fig.5 shows that the proposed DDSRA algorithm achieves a much less FL latency than the baseline schemes, and the advantage of DDSRA algorithm is increasingly obvious as the communication round elapses. Compared with Loss Driven Scheduling, the DDSRA algorithm with  $V = 0.01$  reduces the training latency by 26% for SVHN dataset and 23% for CIFAR-10 dataset, respectively. Compared with Delay Driven Scheduling, the DDSRA algorithm with  $V = 0.01$  prolongs the training latency by 6% for SVHN dataset and 7% for CIFAR-10 dataset, while improving the test accuracy by 7% for SVHN dataset and 17% for CIFAR-10 dataset, respectively.

Fig.6 shows the participation rate comparison between the proposed DDSRA algorithm (with  $V = 0.01, 1000, \text{ and } 10000$ ) and the baselines. First, it can be observed that for CIFAR-10 dataset, the 1-th, 4-th and 5-th gateways and the respective associated devices rarely participate in FL training process in the Loss Driven Scheduling. This is due to that the local datasets at the devices associated with the 1-th, 4-th and 5-th gateways are assigned with a wider variety of the non-IID data points than the other devices. That is, the 1-th, 4-th and 5-th gateways and the respective associated devices are removed from the FL training process by the Loss Driven Scheduling since they achieve a lower training accuracy in each communication round. Second, Delay Driven Scheduling excludes the 4-th gateway and its associated devices from the training process due to long transmission distance to the BS. This reduces the training latency at the cost of degrading the FL performance. Meanwhile, the proposed DDSRA algorithm saves the slow gateways and devices from being excluded from FL training process. To complete the FL training with limited harvested energy, the DDSRA algorithm lowers computational frequency and transmit power for the offloaded local model training and model transmitting, which improves the FL performance but increases the FL training latency. Third, the proposed algorithm achieves a much higher participation rate than the baselines, which contributes to the better learning performance as shown in Fig.4. In addition, it can be observed that a smaller  $V$  encourages more gateways and devices participating in the FL process, which leads to a better FL performance. The experiential results shows that the proposed algorithm can not only save the slow devices from being excluded from FL training process, but also involve important devices in more communication rounds on the track of low latency by setting a larger participation rate for the important devices.

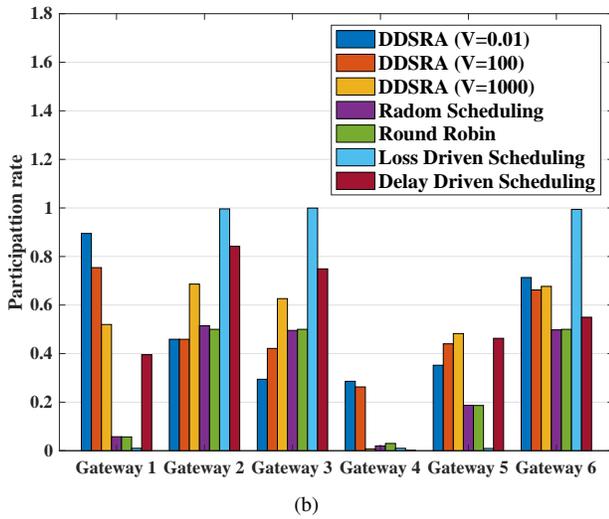
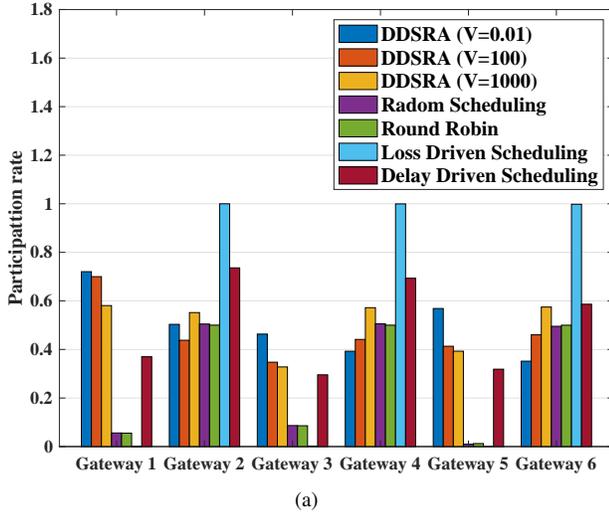


Fig. 6: Participation rate comparison between DDSRA algorithm and the baseline schemes on (a) SVHN and (b) CIFAR-10 datasets.

## VIII. CONCLUSION

In this paper, we develop a communication-computation efficient FL framework for resource-limited IIoT networks that integrates DNN partition technique into the standard FL mechanism. By jointly optimizing the DNN partition point, channel assignment, transmit power, and computation frequency, the proposed DDSRA algorithm can be applied in a wide variety of device heterogeneity scenarios. With the developed device-specific participation rate, the DDSRA algorithm is robust against data heterogeneity by involving more devices with better data distribution over more communication rounds. Thanks to the layer-level memory usage and FLOPs calculation model, the DDSRA algorithm is widely applicable to other large-scale DNN models. Furthermore, we characterize a trade-off of  $[\mathcal{O}(1/V), \mathcal{O}(\sqrt{V})]$  between the FL training latency minimization and the degree of which the participation rate constraint is satisfied with a control parameter  $V$ . The analytical convergence bound shows that the FL convergence rate can be improved by increasing the training data size and setting a higher participation rate for

the important devices with better data distribution. Finally, experimental results demonstrate the developed device-specific participation rate in terms of feasibility. In addition, it has also been shown that DDSRA can obtain higher learning accuracy than the baselines under limited energy supply and memory capacity.

Several interesting directions immediately follow from this work. First, this work utilizes FLOPs to approximate the layer-level training latency and energy consumption. To provide a more accurate estimate, it is of interest to measure the latency and energy consumption by the DNN model training in real-world experiments. Second, due to the feedback loops in hidden layers, how to adapt the proposed FL framework to other large-scale artificial neural networks such as Recurrent Neural Network remains challenging.

## APPENDIX A PROOF OF THEOREM 1

Before we show the main proof of **Theorem 1**, we first give **Lemma 2** below.

**Lemma 2:** For any local epoch  $k$  and communication round  $t$ , we have

$$\left\| \mathbf{w}_n^{k,t} - \mathbf{v}^{k,t} \right\| \leq \frac{\delta_n}{L_n} \left( (\beta L_n + 1)^k - 1 \right), \quad (36)$$

and

$$\mathbb{E} \left\| \tilde{\mathbf{w}}_n^{k,t} - \mathbf{w}_n^{k,t} \right\| \leq \frac{\sigma_n}{L_n \sqrt{\tilde{D}_n}} \left( (\beta L_n + 1)^k - 1 \right). \quad (37)$$

*Proof:* The upper bound of  $\left\| \mathbf{w}_n^{k,t} - \mathbf{v}^{k,t} \right\|$  in (36) is derived by induction. Initially, the upper bound of  $\left\| \mathbf{w}_n^{k,t} - \mathbf{v}^{k,t} \right\|$  in (36) holds when  $k = 0$  since  $\mathbf{w}_n^{0,t} = \mathbf{v}^{0,t}$ . Suppose that (36) holds at the  $k$ -th local epoch. Then, according to the update rule, it can be derived that

$$\begin{aligned} \left\| \mathbf{w}_n^{k+1,t} - \mathbf{v}^{k+1,t} \right\| &= \left\| \mathbf{w}_n^{k,t} - \beta \nabla F_n(\mathbf{w}_n^{k,t}) - \mathbf{v}^{k,t} + \beta \nabla F(\mathbf{v}^{k,t}) \right\| \\ &\leq \left\| \mathbf{w}_n^{k,t} - \mathbf{v}^{k,t} \right\| + \beta \left\| \nabla F_n(\mathbf{w}_n^{k,t}) - \nabla F_n(\mathbf{v}^{k,t}) \right\| + \beta \left\| \nabla F_n(\mathbf{v}^{k,t}) - \nabla F(\mathbf{v}^{k,t}) \right\| \\ &\leq (1 + \beta L_n) \left\| \mathbf{w}_n^{k,t} - \mathbf{v}^{k,t} \right\| + \beta \delta_n \leq \frac{\delta_n}{L_n} \left( (\beta L_n + 1)^{k+1} - 1 \right). \end{aligned} \quad (38)$$

As a result, the upper bound of  $\left\| \mathbf{w}_n^{k,t} - \mathbf{v}^{k,t} \right\|$  in (36) also holds at the  $(k+1)$ -th local epoch. This concludes the proof of (36) in **Lemma 2**.

Note that from **Assumption 1**, it can be derived that  $\mathbb{E} \left\| \nabla \tilde{F}_n(\tilde{\mathbf{w}}_n^{k,t}) - \nabla F_n(\tilde{\mathbf{w}}_n^{k,t}) \right\| \leq \frac{\sigma_n}{\sqrt{\tilde{D}_n}}$ . Similarly, the upper bound of  $\mathbb{E} \left\| \tilde{\mathbf{w}}_n^{k,t} - \mathbf{w}_n^{k,t} \right\|$  can be obtained by induction. ■

Based on **Lemma 2**, it can be derived that

$$\begin{aligned} \left\| \tilde{\mathbf{w}}_m^t - \mathbf{v}^{K,t} \right\| &= \left\| \sum_n \frac{a_{m,n} \tilde{D}_n}{\sum_n a_{m,n} \tilde{D}_n} \tilde{\mathbf{w}}_n^{K,t} - \mathbf{v}^{K,t} \right\| \\ &\leq \sum_n \frac{a_{m,n} \tilde{D}_n}{\sum_n a_{m,n} \tilde{D}_n} \left( \left\| \tilde{\mathbf{w}}_n^{K,t} - \mathbf{w}_n^{K,t} \right\| + \left\| \mathbf{w}_n^{K,t} - \mathbf{v}^{K,t} \right\| \right) \\ &\leq \sum_n \frac{a_{m,n} \tilde{D}_n}{\sum_n a_{m,n} \tilde{D}_n} \left( \frac{\sigma_n}{L_n \sqrt{\tilde{D}_n}} + \frac{\delta_n}{L_n} \right) \left( (\beta L_n + 1)^K - 1 \right). \end{aligned} \quad (39)$$

This concludes the proof of **Theorem 1**.

APPENDIX B  
PROOF OF LEMMA 1

First, from (14), we have

$$Q_m(t+1)^2 \leq Q_m(t)^2 + (\Gamma_m - \mathbb{1}_m^t)^2 + 2Q_m(t)(\Gamma_m - \mathbb{1}_m^t). \quad (40)$$

Next, by moving  $Q_m(t)^2$  to the left-hand side of (40), dividing both sides by 2, summing up the inequalities from  $m = 1$  to  $M$ , and taking the conditional expectation, it can be derived that

$$\begin{aligned} \Delta \Xi(t) &\leq \sum_{m \in \mathcal{M}} \mathbb{E} \{ Q_m(t)(\Gamma_m - \mathbb{1}_m^t) | \mathbf{Q}(t) \} \\ &\quad + \frac{1}{2} \sum_{m \in \mathcal{M}} \mathbb{E} \{ \Gamma_m + \mathbb{1}_m^t | \mathbf{Q}(t) \}. \end{aligned} \quad (41)$$

Note that  $\mathbb{1}_m^t = \sum_{j \in \mathcal{J}} I_{m,j}(t)$ . Given constraints **C1** and **C3**, i.e.,  $I_{m,j}(t) \in \{0, 1\}, \forall m \in \mathcal{M}, j \in \mathcal{J}$ , and  $\sum_{j \in \mathcal{J}} I_{m,j}(t) \leq 1, \forall j \in \mathcal{J}$ , it can be derived that  $0 \leq \mathbb{1}_m^t \leq 1$ . Thus, the upper bound of the conditional Lyapunov drift  $\Delta \Xi(t)$  can be derived as

$$\Delta \Xi(t) \leq \frac{1}{2} \sum_{m \in \mathcal{M}} (\Gamma_m + 1) + \sum_{m \in \mathcal{M}} \mathbb{E} \{ Q_m(t)(\Gamma_m - \mathbb{1}_m^t) | \mathbf{Q}(t) \}. \quad (42)$$

This concludes the proof of **Lemma 1**.

APPENDIX C  
PROOF OF THEOREM 2

Before we represent the main proof of **Theorem 2**, we first give **Lemma 3** below.

**Lemma 3:** For any  $\varsigma > 0$ , there exists an IID policy  $\pi'$  such that

$$\mathbb{E} \{ \tau(t) | \pi' \} \leq \varphi^{\text{opt}} + \varsigma, \quad \mathbb{E} \{ \mathbb{1}_m^t | \pi' \} \geq \Gamma_m - \varsigma. \quad (43)$$

*Proof:* Given any  $\varsigma > 0$ , we can note that there exists a policy  $\pi^0$  which meets all of the constraints in **P0** and yields that  $\lim_{T \rightarrow \infty} \inf \left[ \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} \{ \tau(t) | \pi^0 \} \right] \leq \varphi^{\text{opt}} + \varsigma$ , and  $\lim_{T \rightarrow \infty} \sup \left[ \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} \{ \mathbb{1}_m^t | \pi^0 \} \right] \geq \Gamma_m - \varsigma$ . For a integer  $T_0$ , it can be derived that

$$\frac{1}{T_0} \sum_{t=0}^{T_0-1} \mathbb{E} \{ \tau(t) | \pi^0 \} \leq \varphi^{\text{opt}} + \varsigma, \quad (44)$$

$$\frac{1}{T_0} \sum_{t=0}^{T_0-1} \mathbb{E} \{ \mathbb{1}_m^t | \pi^0 \} \geq \Gamma_m - \varsigma. \quad (45)$$

From [3], we can note that there exists an IID policy  $\pi'$  such that

$$\frac{1}{T_0} \sum_{t=0}^{T_0-1} \mathbb{E} \{ [\tau(t), \mathbb{1}_1^t, \dots, \mathbb{1}_M^t] | \pi^0 \} = \mathbb{E} \{ [\tau(t), \mathbb{1}_1^t, \dots, \mathbb{1}_M^t] | \pi' \}. \quad (46)$$

Thus, by plugging (46) into (44) and (45), we have (43). ■

Next, from **Lemma 1**, we have

$$\Delta_V(t) \leq H + \sum_{m \in \mathcal{M}} \mathbb{E} \{ V\tau(t) + Q_m(t)(\Gamma_m - \mathbb{1}_m^t) | \mathbf{Q}(t), \pi' \}. \quad (47)$$

Plugging (43) into the right-hand-side of (47), letting  $\varsigma \rightarrow 0$ , and taking expectation of both sides, we have

$$\mathbb{E} \{ \Xi(t+1) - \Xi(t) | \mathbf{Q}(t) \} + V \mathbb{E} \{ \tau(t) | \mathbf{Q}(t) \} \leq H + V \varphi^{\text{opt}}. \quad (48)$$

By summing up (48) from  $t = 0$  to  $T - 1$ , and dividing both sides by  $T$  and  $V$ , we have

$$\frac{\sum_{t=1}^T \tau(t)}{T} \leq \varphi^{\text{opt}} + \frac{H}{V} + \frac{\mathbb{E} \{ \Xi(0) - \Xi(T) \}}{VT}, \quad (49)$$

which concludes the proof of (32).

Next, from (48), it can be derived that

$$\Delta \Xi(t) \leq H + V(\varphi^{\text{opt}} - \tau^{\min}), \quad (50)$$

where  $\tau^{\min} = \frac{K \min_{n \in \mathcal{N}} \{ \bar{D}_n \} \sum_{l=1}^L (o_l + o'_l)}{\min \{ \min_{n \in \mathcal{N}} \{ \phi_n^D F_n^D \}, \min_{m \in \mathcal{M}} \{ \phi_m^G F_m^{G, \max} \} \}} + \gamma / B^u / \log_2 \left( 1 + \frac{P_m^{\max} h_{m,j}^u}{(B^u N_0 + I_{m,j}^u)} \right) + \gamma / B^d / \log_2 \left( 1 + \frac{P^B h_{m,j}^d}{(B^d N_0 + I_{m,j}^d)} \right)$ . By summing up (50) from  $t = 0$  to  $T - 1$ , taking expectations, dividing both sides by  $T$ , and recalling that  $\Xi(t) = \frac{1}{2} \sum_{m \in \mathcal{M}} Q_m(t)^2$ , it can be derived that

$$\sum_{m \in \mathcal{M}} \frac{\mathbb{E} \{ Q_m(T)^2 \}}{T} \leq H + V(\varphi^{\text{opt}} - \tau^{\min}) + \sum_{m \in \mathcal{M}} \frac{\mathbb{E} \{ Q_m(0)^2 \}}{T}. \quad (51)$$

Thus, for each gateway and the associated devices, we have

$$\frac{\mathbb{E} \{ Q_m(T)^2 \}}{T} \leq H + V(\varphi^{\text{opt}} - \tau^{\min}) + \sum_{m \in \mathcal{M}} \frac{\mathbb{E} \{ Q_m(0)^2 \}}{T}. \quad (52)$$

By dividing both sides of (52) by  $T$ , and taking the square root of both sides, we have

$$\frac{\mathbb{E} \{ Q_m(T) \}}{T} \leq \sqrt{\frac{H + V(\varphi^{\text{opt}} - \tau^{\min})}{T} + \sum_{m \in \mathcal{M}} \frac{\mathbb{E} \{ Q_m(0)^2 \}}{T^2}}. \quad (53)$$

From (14), it can be derived that

$$Q_m(t+1) \geq Q_m(t) - \mathbb{1}_m^t + \Gamma_m. \quad (54)$$

Note that  $\mathbb{E} \{ Q_m(0) \} < \infty$ . By summing up (54) from  $t = 0$  to  $T - 1$ , taking expectations, and dividing both sides by  $T$ , it can be derived that  $\frac{\mathbb{E} \{ Q_m(T) \}}{T} \geq \Gamma_m - \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{1}_m^t$ . Thus, from (53), we have

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{1}_m^t \geq \Gamma_m - \sqrt{\frac{H + V(\varphi^{\text{opt}} - \tau^{\min})}{T} + \sum_{m \in \mathcal{M}} \frac{\mathbb{E} \{ Q_m(0)^2 \}}{T^2}}, \quad (55)$$

This concludes the proof of (33).

APPENDIX D  
PROOF OF THEOREM 3

Before we show the main proof of **Theorem 3**, we first give **Lemma 4** below.

**Lemma 4:** For any local epoch  $k$  and communication round  $t$ , we have

$$\| \mathbf{w}^{k,t} - \mathbf{v}^{k,t} \| \leq \frac{\delta}{L} \left( (\beta L + 1)^k - 1 \right) - \beta \delta k, \quad (56)$$

$$\begin{aligned} \mathbb{E} \| \tilde{\mathbf{w}}^{k,t} - \mathbf{w}^{k,t} \| &\leq \frac{1}{L} \left( \sum_{n \in \mathcal{N}} \xi_n \frac{\sigma_n}{\sqrt{D_n(t)}} \right) \left( (\beta L + 1)^k - 1 \right) \\ &\quad + \beta k \left( \sum_{n \in \mathcal{N}} \left| \xi_n - \frac{D_n}{\sum_{n \in \mathcal{N}} D_n} \right| \rho_n \right), \end{aligned} \quad (57)$$

where  $\xi_n = \frac{\sum_{m \in \mathcal{M}} \Gamma_m a_{m,n} \bar{D}_n}{\sum_{n \in \mathcal{N}} \sum_{m \in \mathcal{M}} \Gamma_m a_{m,n} \bar{D}_n}$ .

*Proof:* In this proof, we first derive the upper bound of  $\| \mathbf{w}^{k,t} - \mathbf{v}^{k,t} \|$  in (56) by induction. Initially, the upper bound of  $\| \mathbf{w}^{k,t} - \mathbf{v}^{k,t} \|$  in (56) holds at  $k = 0$  since  $\mathbf{w}^{0,t} = \mathbf{v}^{0,t}$ . Suppose that (56) holds at the  $k$ -th local epoch. According to the update rule, it can be derived that

$$\begin{aligned} \| \mathbf{w}^{k+1,t} - \mathbf{v}^{k+1,t} \| &= \left\| \mathbf{w}^{k,t} - \mathbf{v}^{k,t} - \frac{\beta}{\sum_n D_n} \sum_n D_n \left( \nabla F_n(\mathbf{w}_n^{k,t}) \right. \right. \\ &\quad \left. \left. - \nabla F_n(\mathbf{v}_n^{k,t}) \right) \right\| \leq \| \mathbf{w}^{k,t} - \mathbf{v}^{k,t} \| + \frac{\beta}{\sum_n D_n} \sum_n D_n \| \nabla F_n(\mathbf{w}_n^{k,t}) \end{aligned}$$

$$\begin{aligned}
& -\nabla F_n(\mathbf{v}^{k,t}) \Big\| \leq \left\| \mathbf{w}^{k,t} - \mathbf{v}^{k,t} \right\| + \frac{\beta L}{\sum_n D_n} \sum_n D_n \left\| \mathbf{w}_n^{k,t} - \mathbf{v}^{k,t} \right\| \\
& \leq \left\| \mathbf{w}^{k,t} - \mathbf{v}^{k,t} \right\| + \beta \delta (\beta L + 1)^k - \beta \delta \leq \frac{\delta}{L} \left( (\beta L + 1)^{k+1} - 1 \right) \\
& - \beta \delta (k + 1). \tag{58}
\end{aligned}$$

As a result, the upper bound of  $\left\| \mathbf{w}^{k,t} - \mathbf{v}^{k,t} \right\|$  in (56) also holds at the  $(k + 1)$ -th local epoch. This concludes the proof of (56) in **Lemma 4**.

Next, we obtain the upper bound of  $\mathbb{E} \left\| \tilde{\mathbf{w}}^{k,t} - \mathbf{w}^{k,t} \right\|$  by induction as follows. Initially, the upper bound of  $\mathbb{E} \left\| \tilde{\mathbf{w}}_n^{k,t} - \mathbf{w}_n^{k,t} \right\|$  holds at  $k = 0$  since  $\tilde{\mathbf{w}}_n^{0,t} = \mathbf{w}_n^{0,t}$ . Suppose that the upper bound of  $\mathbb{E} \left\| \tilde{\mathbf{w}}_n^{k,t} - \mathbf{w}_n^{k,t} \right\|$  in (37) holds at the  $k$ -th local epoch. Thus, it can be derived that

$$\begin{aligned}
& \left\| \tilde{\mathbf{w}}^{k+1,t} - \mathbf{w}^{k+1,t} \right\| = \left\| \tilde{\mathbf{w}}^{k,t} - \beta \sum_n \frac{\sum_m \mathbb{1}_m^t a_{m,n} \tilde{D}_n}{\sum_n \sum_m \mathbb{1}_m^t a_{m,n} \tilde{D}_n} \nabla \tilde{F}_n \left( \tilde{\mathbf{w}}_n^{k,t} \right) - \mathbf{w}^{k,t} + \beta \sum_n \frac{D_n}{\sum_n D_n} \nabla F_n \left( \mathbf{w}_n^{k,t} \right) \right\| \leq \left\| \tilde{\mathbf{w}}^{k,t} - \mathbf{w}^{k,t} \right\| + \beta \\
& \left\| \sum_n \frac{\sum_m \mathbb{1}_m^t a_{m,n} \tilde{D}_n}{\sum_n \sum_m \mathbb{1}_m^t a_{m,n} \tilde{D}_n} \nabla \tilde{F}_n \left( \tilde{\mathbf{w}}_n^{k,t} \right) - \sum_n \frac{D_n}{\sum_n D_n} \nabla F_n \left( \mathbf{w}_n^{k,t} \right) \right\| \\
& \leq \left\| \tilde{\mathbf{w}}^{k,t} - \mathbf{w}^{k,t} \right\| + \beta \sum_n \left\| \left( \nabla \tilde{F}_n \left( \tilde{\mathbf{w}}_n^{k,t} \right) - \nabla F_n \left( \tilde{\mathbf{w}}_n^{k,t} \right) + \nabla F_n \left( \tilde{\mathbf{w}}_n^{k,t} \right) - \nabla F_n \left( \mathbf{w}_n^{k,t} \right) \right) \frac{\sum_m \mathbb{1}_m^t a_{m,n} \tilde{D}_n}{\sum_n \sum_m \mathbb{1}_m^t a_{m,n} \tilde{D}_n} \right\| + \beta \sum_n \left\| \nabla F_n \left( \mathbf{w}_n^{k,t} \right) \left( \frac{\sum_m \mathbb{1}_m^t a_{m,n} \tilde{D}_n}{\sum_n \sum_m \mathbb{1}_m^t a_{m,n} \tilde{D}_n} - \frac{D_n}{\sum_n D_n} \right) \right\| \leq \left\| \tilde{\mathbf{w}}^{k,t} - \mathbf{w}^{k,t} \right\| + \beta \sum_n \left\| \frac{\sum_m \mathbb{1}_m^t a_{m,n} \tilde{D}_n}{\sum_n \sum_m \mathbb{1}_m^t a_{m,n} \tilde{D}_n} \left( \nabla \tilde{F}_n \left( \tilde{\mathbf{w}}_n^{k,t} \right) - \nabla F_n \left( \tilde{\mathbf{w}}_n^{k,t} \right) \right) \right\| + \beta \left\| \nabla F_n \left( \tilde{\mathbf{w}}_n^{k,t} \right) - \nabla F_n \left( \mathbf{w}_n^{k,t} \right) \right\| \frac{\sum_m \mathbb{1}_m^t a_{m,n} \tilde{D}_n}{\sum_n \sum_m \mathbb{1}_m^t a_{m,n} \tilde{D}_n} \right\| + \beta \sum_n \left\| \nabla F_n \left( \mathbf{w}_n^{k,t} \right) \left( \frac{\sum_m \mathbb{1}_m^t a_{m,n} \tilde{D}_n}{\sum_n \sum_m \mathbb{1}_m^t a_{m,n} \tilde{D}_n} - \frac{D_n}{\sum_n D_n} \right) \right\| \leq \left\| \tilde{\mathbf{w}}^{k,t} - \mathbf{w}^{k,t} \right\| + \beta \sum_n \left\| \frac{\sum_m \mathbb{1}_m^t a_{m,n} \tilde{D}_n}{\sum_n \sum_m \mathbb{1}_m^t a_{m,n} \tilde{D}_n} \left\| \left( \nabla \tilde{F}_n \left( \tilde{\mathbf{w}}_n^{k,t} \right) - \nabla F_n \left( \tilde{\mathbf{w}}_n^{k,t} \right) \right) \right\| + \left\| \nabla F_n \left( \tilde{\mathbf{w}}_n^{k,t} \right) - \nabla F_n \left( \mathbf{w}_n^{k,t} \right) \right\| \right\| + \beta \sum_n \left\| \frac{\sum_m \mathbb{1}_m^t a_{m,n} \tilde{D}_n}{\sum_n \sum_m \mathbb{1}_m^t a_{m,n} \tilde{D}_n} - \frac{D_n}{\sum_n D_n} \right\| \left\| \nabla F_n \left( \mathbf{w}_n^{k,t} \right) \right\|. \tag{59}
\end{aligned}$$

Based on (37) in **Lemma 2** and **Assumption 1**, we have

$$\begin{aligned}
& \left\| \tilde{\mathbf{w}}^{k+1,t} - \mathbf{w}^{k+1,t} \right\| \leq \left\| \tilde{\mathbf{w}}^{k,t} - \mathbf{w}^{k,t} \right\| + \beta \sum_n \left( \frac{\sigma_n}{\sqrt{\tilde{D}_n}} + \frac{\sigma_n}{\sqrt{\tilde{D}_n}} \left( (\beta L_n + 1)^k - 1 \right) \right) \left\| \frac{\sum_m \mathbb{1}_m^t a_{m,n} \tilde{D}_n}{\sum_n \sum_m \mathbb{1}_m^t a_{m,n} \tilde{D}_n} \right\| + \beta \sum_n \left\| -\frac{D_n}{\sum_n D_n} + \frac{\sum_m \mathbb{1}_m^t a_{m,n} \tilde{D}_n}{\sum_n \sum_m \mathbb{1}_m^t a_{m,n} \tilde{D}_n} \right\| \rho_n. \tag{60}
\end{aligned}$$

By taking expectation of both sides, we have

$$\begin{aligned}
& \mathbb{E} \left\| \tilde{\mathbf{w}}^{k+1,t} - \mathbf{w}^{k+1,t} \right\| \leq \mathbb{E} \left\| \tilde{\mathbf{w}}^{k,t} - \mathbf{w}^{k,t} \right\| + \beta \sum_n \|\xi_n\| \frac{\sigma_n}{\sqrt{\tilde{D}_n}} \\
& (\beta L_n + 1)^k + \beta \sum_n \left\| \xi_n - \frac{D_n}{\sum_n D_n} \right\| \rho_n. \tag{61}
\end{aligned}$$

Plugging (57) in **Lemma 4** into the right-hand-side of (61), it can be proved that the upper bound of  $\mathbb{E} \left\| \tilde{\mathbf{w}}^{k,t} - \mathbf{w}^{k,t} \right\|$  in (57) also holds at the  $(k + 1)$ -th local epoch. This concludes the proof of (57) in **Lemma 4**.  $\blacksquare$

Thus, based on **Lemma 4**, it can be derived that

$$\begin{aligned}
\mathbb{E} \left\| \tilde{\mathbf{w}}^{k,t} - \mathbf{w}^{k,t} \right\| & \leq \frac{1}{L} \left( \delta + \sum_n \xi_n \frac{\sigma_n}{\sqrt{\tilde{D}_n}} \right) \left( (\beta L + 1)^k - 1 \right) \\
& + \beta k \left( \delta + \sum_n \left| \xi_n - \frac{D_n}{\sum_n D_n} \right| \rho_n \right). \tag{62}
\end{aligned}$$

Based on (62), the detailed proof of **Theorem 3** can be found in [32].

## APPENDIX E PROOF OF THEOREM 4

First, due to  $F(\mathbf{w})$  is  $L$ -smooth, it can be derived that

$$\begin{aligned}
& F(\tilde{\mathbf{w}}^{t+1}) - F(\tilde{\mathbf{w}}^t) \\
& \leq \frac{L}{2} \left\| \tilde{\mathbf{w}}^{t+1} - \tilde{\mathbf{w}}^t \right\|^2 + \langle \nabla F(\tilde{\mathbf{w}}^t), \tilde{\mathbf{w}}^{t+1} - \tilde{\mathbf{w}}^t \rangle. \tag{63}
\end{aligned}$$

Taking the conditional expectation on both sides of (63), we have

$$\begin{aligned}
& \mathbb{E} [F(\tilde{\mathbf{w}}^{t+1}) | \tilde{\mathbf{w}}^t] - F(\tilde{\mathbf{w}}^t) \leq \frac{L}{2} \mathbb{E} \left[ \left\| \tilde{\mathbf{w}}^{t+1} - \tilde{\mathbf{w}}^t \right\|^2 | \tilde{\mathbf{w}}^t \right] \\
& + \langle \nabla F(\tilde{\mathbf{w}}^t), \mathbb{E} [\tilde{\mathbf{w}}^{t+1} - \tilde{\mathbf{w}}^t | \tilde{\mathbf{w}}^t] \rangle. \tag{64}
\end{aligned}$$

Second, according to the update rule, we have

$$\begin{aligned}
& \mathbb{E} [F(\tilde{\mathbf{w}}^{t+1}) | \tilde{\mathbf{w}}^t] - F(\tilde{\mathbf{w}}^t) = \frac{L\beta^2}{2} \mathbb{E} \left[ \left\| \sum_n \frac{\sum_m \mathbb{1}_m^t a_{m,n} \tilde{D}_n}{\sum_n \sum_m \mathbb{1}_m^t a_{m,n} \tilde{D}_n} \nabla F_n(\tilde{\mathbf{w}}_n^{k,t}) \right\|^2 | \tilde{\mathbf{w}}^t \right] + \left\langle \nabla F(\tilde{\mathbf{w}}^t), \mathbb{E} \left[ -\beta \sum_{n=k=0}^{K-1} \nabla F_n(\tilde{\mathbf{w}}_n^{k,t}) \frac{\sum_m \mathbb{1}_m^t a_{m,n} \tilde{D}_n}{\sum_n \sum_m \mathbb{1}_m^t a_{m,n} \tilde{D}_n} \middle| \tilde{\mathbf{w}}^t \right] \right\rangle \leq \frac{L\beta^2 NK}{2} \mathbb{E} \left[ \sum_n \sum_{k=0}^{K-1} \left\| \nabla F_n(\tilde{\mathbf{w}}_n^{k,t}) \frac{\sum_m \mathbb{1}_m^t a_{m,n} \tilde{D}_n}{\sum_n \sum_m \mathbb{1}_m^t a_{m,n} \tilde{D}_n} \right\|^2 | \tilde{\mathbf{w}}^t \right] + \left\langle \mathbb{E} \left[ -\beta \sum_n \frac{\sum_m \mathbb{1}_m^t a_{m,n} \tilde{D}_n}{\sum_n \sum_m \mathbb{1}_m^t a_{m,n} \tilde{D}_n} \sum_{k=0}^{K-1} \nabla F_n(\tilde{\mathbf{w}}_n^{k,t}) \middle| \tilde{\mathbf{w}}^t \right], \nabla F(\tilde{\mathbf{w}}^t) \right\rangle = \frac{L\beta^2 NK}{2} \sum_n \sum_{k=0}^{K-1} \xi_n^2 \mathbb{E} \left[ \left\| \nabla F_n(\tilde{\mathbf{w}}_n^{k,t}) \right\|^2 | \tilde{\mathbf{w}}^t \right] + \left\langle \nabla F(\tilde{\mathbf{w}}^t), -\beta \sum_n \sum_{k=0}^{K-1} \xi_n \mathbb{E} \left[ \nabla F_n(\tilde{\mathbf{w}}_n^{k,t}) \middle| \tilde{\mathbf{w}}^t \right] \right\rangle. \tag{65}
\end{aligned}$$

Note that  $\xi_n = \frac{\sum_{m \in \mathcal{M}} \Gamma_m a_{m,n} \tilde{D}_n}{\sum_{n \in \mathcal{N}} \sum_{m \in \mathcal{M}} \Gamma_m a_{m,n} \tilde{D}_n}$ . Taking the expectation on both sides of (65), we have

$$\begin{aligned}
& \mathbb{E} [F(\tilde{\mathbf{w}}^{t+1})] - \mathbb{E} [F(\tilde{\mathbf{w}}^t)] \leq \frac{L\beta^2 NK}{2} \sum_n \sum_{k=0}^{K-1} \xi_n^2 \mathbb{E} \left[ \left\| \nabla F_n(\tilde{\mathbf{w}}_n^{k,t}) \right\|^2 \right] + \left\langle \mathbb{E} [\nabla F(\tilde{\mathbf{w}}^t)], -\beta \sum_n \sum_{k=0}^{K-1} \xi_n \mathbb{E} \left[ \nabla F_n(\tilde{\mathbf{w}}_n^{k,t}) \right] \right\rangle = \frac{L\beta^2 NK}{2} \\
& \sum_n \sum_{k=0}^{K-1} \xi_n^2 \mathbb{E} \left[ \left\| \nabla F_n(\tilde{\mathbf{w}}_n^{k,t}) \right\|^2 \right] + \beta \sum_{k=0}^{K-1} \mathbb{E} \left[ \left\langle -\sum_n \xi_n \nabla F_n(\tilde{\mathbf{w}}_n^{k,t}), \nabla F(\tilde{\mathbf{w}}^t) \right\rangle \right]. \tag{66}
\end{aligned}$$

Note that

$$\mathbb{E} \left[ \left\langle \nabla F(\tilde{\mathbf{w}}^t), -\sum_n \xi_n \nabla F_n(\tilde{\mathbf{w}}_n^{k,t}) \right\rangle \right] = \mathbb{E} \left[ \left\langle \nabla F(\tilde{\mathbf{w}}^t), \nabla F(\tilde{\mathbf{w}}^t) \right\rangle \right]$$

$$\begin{aligned}
& -\nabla F(\tilde{\mathbf{w}}^t) - \sum_n \xi_n \nabla F_n(\tilde{\mathbf{w}}_n^{k,t}) \Bigg\rangle = -\mathbb{E} \left[ \langle \nabla F(\tilde{\mathbf{w}}^t), \nabla F(\tilde{\mathbf{w}}^t) \rangle \right] \\
& + \mathbb{E} \left[ \left\langle \nabla F(\tilde{\mathbf{w}}^t), \nabla F(\tilde{\mathbf{w}}^t) - \sum_n \xi_n \nabla F_n(\tilde{\mathbf{w}}_n^{k,t}) \right\rangle \right] \leq -\frac{1}{2} \mathbb{E} \left[ \left\| \nabla F(\tilde{\mathbf{w}}^t) \right\|^2 \right] \\
& + \frac{1}{2} \mathbb{E} \left[ \left\| \nabla F(\tilde{\mathbf{w}}^t) - \sum_n \xi_n \nabla F_n(\tilde{\mathbf{w}}_n^{k,t}) \right\|^2 \right] \leq -\frac{1}{2} \mathbb{E} \left[ \left\| \nabla F(\tilde{\mathbf{w}}^t) \right\|^2 \right] \\
& + \frac{N}{2} \sum_n \xi_n^2 \mathbb{E} \left[ \left\| \nabla F(\tilde{\mathbf{w}}^t) - \nabla F_n(\tilde{\mathbf{w}}_n^{k,t}) \right\|^2 \right] \leq -\frac{1}{2} \mathbb{E} \left[ \left\| \nabla F(\tilde{\mathbf{w}}^t) \right\|^2 \right] \\
& + \frac{N}{2} \sum_n \xi_n^2 L_n^2 \mathbb{E} \left[ \left\| \tilde{\mathbf{w}}^t - \tilde{\mathbf{w}}_n^{k,t} \right\|^2 \right] \leq -\frac{1}{2} \mathbb{E} \left[ \left\| \nabla F(\tilde{\mathbf{w}}^t) \right\|^2 \right] \\
& + \frac{N}{2} \sum_n \xi_n^2 L_n^2 \beta^2 \mathbb{E} \left[ \left\| \sum_{j=0}^{k-1} \nabla F_n(\tilde{\mathbf{w}}_n^{j,t}) \right\|^2 \right] \leq -\frac{1}{2} \mathbb{E} \left[ \left\| \nabla F(\tilde{\mathbf{w}}^t) \right\|^2 \right] \\
& + \frac{N}{2} \sum_n \xi_n^2 L_n^2 \beta^2 k \sum_{j=0}^{k-1} \mathbb{E} \left[ \left\| \nabla F_n(\tilde{\mathbf{w}}_n^{j,t}) \right\|^2 \right]. \quad (67)
\end{aligned}$$

Plugging (67) into the right-hand-side of (66), we have

$$\begin{aligned}
\mathbb{E} [F(\tilde{\mathbf{w}}^{t+1})] - \mathbb{E} [F(\tilde{\mathbf{w}}^t)] & \leq \frac{L\beta^2 NK}{2} \sum_n \sum_{k=0}^{K-1} \xi_n^2 \mathbb{E} \left[ \left\| \nabla F_n(\tilde{\mathbf{w}}_n^{k,t}) \right\|^2 \right] \\
& - \frac{K\beta}{2} \mathbb{E} \left[ \left\| \nabla F(\tilde{\mathbf{w}}^t) \right\|^2 \right] + \frac{N\beta^3}{2} \sum_n \sum_{k=0}^{K-1} \xi_n^2 L_n^2 \beta^2 k \sum_{j=0}^{k-1} \mathbb{E} \left[ \left\| \nabla F_n(\tilde{\mathbf{w}}_n^{j,t}) \right\|^2 \right]. \quad (68)
\end{aligned}$$

Finally, by summing up (68) form  $t = 0$  to  $T - 1$ , we have

$$\begin{aligned}
\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} \left[ \left\| \nabla F(\tilde{\mathbf{w}}^t) \right\|^2 \right] & \leq \frac{2}{K\beta T} \left( \mathbb{E} [F(\tilde{\mathbf{w}}^0)] - \mathbb{E} [F(\tilde{\mathbf{w}}^T)] \right) \\
& + \frac{L\beta N}{T} \sum_{t=0}^{T-1} \sum_n \sum_{k=0}^{K-1} \xi_n^2 \mathbb{E} \left[ \left\| \nabla F_n(\tilde{\mathbf{w}}_n^{k,t}) \right\|^2 \right] + \frac{N\beta^2}{KT} \sum_{t=0}^{T-1} \sum_n \sum_{k=0}^{K-1} \xi_n^2 L_n^2 \beta^2 k \sum_{j=0}^{k-1} \mathbb{E} \left[ \left\| \nabla F_n(\tilde{\mathbf{w}}_n^{j,t}) \right\|^2 \right]. \quad (69)
\end{aligned}$$

This completes the proof of **Theorem 4**.

## REFERENCES

- [1] J. Zhou, Q. Lu, W. Dai, and E. Herrera-Viedma, "Guest Editorial: Federated learning for industrial IoT in Industry 4.0," *IEEE Trans. Ind. Informatics*, vol. 17, no. 12, pp. 8438–8441, 2021.
- [2] Y. Yang, "Multi-tier computing networks for intelligent IoT," *Nature Electronics*, vol. 2, no. 3, pp. 4–5, 2019.
- [3] X. Deng, J. Li, L. Shi, Z. Wang, J. H. Wang, and T. Wang, "On dynamic resource allocation for blockchain assisted federated learning over wireless channels," in *Proc. IEEE CPSCOM, Melbourne, Australia, Dec. 2021*, pp. 306–313.
- [4] H. Zhou, C. She, Y. Deng, M. Dohler, and A. Nallanathan, "Machine learning for massive industrial internet of things," *IEEE Wirel. Commun.*, vol. 28, no. 4, pp. 81–87, 2021.
- [5] W. Zhang, D. Yang, W. Wu, H. Peng, N. Zhang, H. Zhang, and X. Shen, "Optimizing federated learning in distributed industrial IoT: A multi-agent approach," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3688–3703, 2021.
- [6] E. Sisinni and A. Mahmood, "Wireless communications for industrial internet of things: The LPWAN solutions," in *Wireless Networks and Industrial IoT*, Springer, 2021, pp. 79–103.
- [7] M. M. Wadu, S. Samarakoon, and M. Bennis, "Joint client scheduling and resource allocation under channel uncertainty in federated learning," *IEEE Trans. Commun.*, vol. 69, no. 9, pp. 5962–5974, 2021.
- [8] W. Zhang, D. Yang, W. Wu, H. Peng, H. Zhang, and X. S. Shen, "Spectrum and computing resource management for federated learning in distributed industrial IoT," in *Proc. IEEE ICC, Montreal, QC, Canada, Jun. 2021*, pp. 1–6.
- [9] W. Gao, Z. Zhao, G. Min, Q. Ni, and Y. Jiang, "Resource allocation for latency-aware federated learning in industrial internet of things," *IEEE Trans. Ind. Informatics*, vol. 17, no. 12, pp. 8505–8513, 2021.

- [10] J. Xu, H. Wang, and L. Chen, "Bandwidth allocation for multiple federated learning services in wireless edge networks," *IEEE Trans. Wirel. Commun.*, vol. 21, no. 4, pp. 2534–2546, 2022.
- [11] S. Liu, G. Yu, X. Chen, and M. Bennis, "Joint user association and resource allocation for wireless hierarchical federated learning with IID and Non-IID data," *IEEE Trans. Wirel. Commun.*, pp. 1–1, 2022.
- [12] D. Chen, C. S. Hong, L. Wang, Y. Zha, Y. Zhang, X. Liu, and Z. Han, "Matching-theory-based low-latency scheme for multitask federated learning in MEC networks," *IEEE Internet Things J.*, vol. 8, no. 14, pp. 11415–11426, 2021.
- [13] Q.-V. Pham, M. Zeng, R. Ruby, T. Huynh-The, and W. Hwang, "UAV communications for sustainable federated learning," *IEEE Trans. Veh. Technol.*, vol. 70, no. 4, pp. 3944–3948, 2021.
- [14] Q.-V. Pham, M. Le, T. Huynh-The, Z. Han, and W.-J. Hwang, "Energy-efficient federated learning over UAV-enabled wireless powered communications," *IEEE Trans. Veh. Technol.*, pp. 1–1, 2022.
- [15] L. Yu, R. Albelaihi, X. Sun, N. Ansari, and M. Devetsikiotis, "Jointly optimizing client selection and resource management in wireless federated learning for internet of things," *IEEE Internet Things J.*, vol. 9, no. 6, pp. 4385–4395, 2022.
- [16] H. Xiao, J. Zhao, Q. Pei, J. Feng, L. Liu, and W. Shi, "Vehicle selection and resource optimization for federated learning in vehicular edge computing," *IEEE Trans. Intell. Transport. Syst.*, pp. 1–15, 2021.
- [17] J. Ren, J. Sun, H. Tian, W. Ni, G. Nie, and Y. Wang, "Joint resource allocation for efficient federated learning in internet of things supported by edge computing," in *Proc. IEEE ICC Workshops, Montreal, QC, Canada, Jun. 2021*, pp. 1–6.
- [18] X. Liu, W. Yu, F. Liang, D. Griffith, and N. Golmie, "Toward deep transfer learning in industrial internet of things," *IEEE Internet Things J.*, vol. 8, no. 15, pp. 12163–12175, 2021.
- [19] J. Zhang, J. Wang, Y. Zhao, and B. Chen, "An efficient federated learning scheme with differential privacy in mobile edge computing," in *Proc. MLCOM, Nanjing, China, Aug. 2019*, pp. 538–550.
- [20] G. V. Demirci and H. Ferhatosmanoglu, "Partitioning sparse deep neural networks for scalable training and inference," in *Proc. ACM ICS, Virtual Event, USA, Jun. 2021*, pp. 254–265.
- [21] J. Zhang, Y. Zhao, J. Wang, and B. Chen, "Fedmec: Improving efficiency of differentially private federated learning via mobile edge computing," *Mob. Networks Appl.*, vol. 25, no. 6, pp. 2421–2433, 2020.
- [22] T. Mohammed, C. Joe-Wong, R. Babbar, and M. D. Francesco, "Distributed inference acceleration with adaptive DNN partitioning and offloading," in *Proc. IEEE INFOCOM, Toronto, ON, Canada, Jul. 2020*, pp. 854–863.
- [23] M. Gao, R. Shen, L. Shi, W. Qi, J. Li, and Y. Li, "Task partitioning and offloading in dnn-task enabled mobile edge computing networks," *IEEE Trans. Mobile Comput.*, pp. 1–1, 2021.
- [24] K. Wei, J. Li, M. Ding, C. Ma, H. Su, B. Zhang, and H. V. Poor, "User-level privacy-preserving federated learning: Analysis and performance optimization," *IEEE Trans. Mobile Comput.*, pp. 1–1, 2021.
- [25] N. Benvenuto and F. Piazza, "On the complex backpropagation algorithm," *IEEE Trans. Signal Process.*, vol. 40, no. 4, pp. 967–969, 1992.
- [26] H. H. Yang, Z. Liu, T. Q. S. Quek, and H. V. Poor, "Scheduling policies for federated learning in wireless networks," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 317–333, 2020.
- [27] K. Siu, D. M. Stuart, M. Mahmoud, and A. Moshovos, "Memory requirements for convolutional neural network hardware accelerators," in *Proc. IEEE IISWC, Raleigh, NC, USA, Sep. 2018*, pp. 111–121.
- [28] D. Justus, J. Brennan, S. Bonner, and A. S. McGough, "Predicting the computational cost of deep learning models," in *Proc. IEEE BigData, Seattle, WA, USA, Dec. 2018*, pp. 3873–3882.
- [29] S. Wu, C. Chakrabarti, and H. Lee, "Reducing energy of baseband processor for IoT terminals with long range wireless communications," *J. Signal Process. Syst.*, vol. 90, no. 10, pp. 1345–1355, 2018.
- [30] C. Chaccour, M. N. Soorki, W. Saad, M. Bennis, and P. Popovski, "Can terahertz provide high-rate reliable low latency communications for wireless VR?" *IEEE Internet Things J.*, pp. 1–1, 2022.
- [31] M. Sikimić, M. Amović, V. Vujović, B. Suknović, and D. Manjak, "An overview of wireless technologies for IoT network," in *Proc. IEEE INFOTEH, East Sarajevo, Bosnia and Herzegovina, Mar. 2020*, pp. 1–6.
- [32] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, 2019.
- [33] X. Lyu, C. Ren, W. Ni, H. Tian, R. P. Liu, and E. Dutkiewicz, "Optimal online data partitioning for geo-distributed machine learning in edge of wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2393–2406, 2019.

- [34] W. Xia, T. Q. S. Quek, K. Guo, W. Wen, H. H. Yang, and H. Zhu, "Multi-armed bandit-based client scheduling for federated learning," *IEEE Trans. Wirel. Commun.*, vol. 19, no. 11, pp. 7108–7123, 2020.
- [35] T. Huang, W. Lin, W. Wu, L. He, K. Li, and A. Y. Zomaya, "An efficiency-boosting client selection scheme for federated learning with fairness guarantee," *IEEE Trans. Parallel Distributed Syst.*, vol. 32, no. 7, pp. 1552–1564, 2021.
- [36] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*, ser. Synthesis Lectures on Communication Networks. Morgan & Claypool Publishers, 2010.
- [37] X. Deng, J. Li, L. Shi, Z. Wei, X. Zhou, and J. Yuan, "Wireless powered mobile edge computing: Dynamic resource allocation and throughput maximization," *IEEE Trans. on Mobile Comput.*, vol. 21, no. 6, pp. 2271–2288, 2022.
- [38] Y. Feng and D. P. Palomar, "SCRIP: Successive convex optimization methods for risk parity portfolio design," *IEEE Trans. Signal Process.*, vol. 63, no. 19, pp. 5285–5300, 2015.
- [39] G. K. Y. Ho, Y. Fang, and B. M. H. Pong, "A multiphysics design and optimization method for air-core planar transformers in high-frequency LLC resonant converters," *IEEE Trans. Ind. Electron.*, vol. 67, no. 2, pp. 1605–1614, 2020.
- [40] R. E. Burkard, M. Dell'Amico, and S. Martello, *Assignment Problems*. SIAM, 2009.
- [41] A. Frank, "On Kuhn's Hungarian Method—A tribute from Hungary," *Naval Research Logistics*, vol. 52, no. 1, pp. 2–5, 2005.
- [42] Y. Fu, H. Mei, K. Wang, and K. Yang, "Joint optimization of 3D trajectory and scheduling for solar-powered UAV systems," *IEEE Trans. Veh. Technol.*, vol. 70, no. 4, pp. 3972–3977, 2021.
- [43] M. Hua, Y. Wang, Q. Wu, H. Dai, Y. Huang, and L. Yang, "Energy-efficient cooperative secure transmission in multi-UAV-enabled wireless networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7761–7775, 2019.
- [44] Q. Wu, Y. Zeng, and R. Zhang, "Joint trajectory and communication design for multi-UAV enabled wireless networks," *IEEE Trans. Wirel. Commun.*, vol. 17, no. 3, pp. 2109–2121, 2018.
- [45] A. T. Phillips, "Quadratic fractional programming: Dinkelbach method," in *Encyclopedia of Optimization*. Springer US, 2001, pp. 2107–2110.
- [46] K. Shen and W. Yu, "Fractional programming for communication systems—Part I: Power control and beamforming," *IEEE Trans. Signal Process.*, vol. 66, no. 10, pp. 2616–2630, 2018.
- [47] P. Sermanet, S. Chintala, and Y. LeCun, "Convolutional neural networks applied to house numbers digit classification," in *Proc. IEEE ICPR*, 2012, pp. 3288–3291.
- [48] L. Yang, D. Bankman, B. Moons, M. Verhelst, and B. Murmann, "Bit error tolerance of a CIFAR-10 binarized convolutional neural network processor," in *Proc. IEEE ISCAS*, 2018, pp. 1–5.
- [49] S. Sharma and S. Singh, "Vision-based hand gesture recognition using deep learning for the interpretation of sign language," *Expert Syst. Appl.*, vol. 182, p. 115657, 2021.
- [50] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with Non-IID data," *CoRR*, vol. abs/1806.00582, 2018.
- [51] R. Dolbeau, "Theoretical peak FLOPS per instruction set: A tutorial," *J. Supercomput.*, vol. 74, no. 3, pp. 1341–1377, 2018.