

# Optimizing Federated Learning Efficiency via Slimmable Neural Networks and AirComp for Resource-Heterogeneous Devices

Yue Zhang, *Graduate Student Member, IEEE*, Guopeng Zhang, Kezhi Wang, *Senior Member, IEEE*,  
and Kun Yang, *Fellow, IEEE*

**Abstract**—With the rapid growth of the Internet of Things (IoT) and edge intelligence, federated learning (FL) has become a promising approach for distributed model training while preserving privacy. However, the resource constraints of IoT devices, including limited communication bandwidth and heterogeneous computational and energy capabilities, hinder the scalability and performance of FL. To address these challenges, we propose the **Air-FedSNN** framework, which integrates slimmable neural networks (SNNs) with over-the-air computation (AirComp) to improve communication, computation, and energy efficiency in heterogeneous IoT networks. A width-aware aggregation mechanism is developed to mitigate the impact of varying model widths and aggregation errors. Additionally, a dynamic width allocation strategy based on Lyapunov optimization and multi-armed bandit (MAB) theory is introduced to adaptively select model widths under long-term energy constraints, balancing energy efficiency and training performance without prior knowledge of device capabilities. Extensive experiments on MNIST, CIFAR-10 and CIFAR-100 datasets demonstrate that **Air-FedSNN** outperforms conventional FL methods, achieving superior accuracy-energy tradeoffs, and provides an efficient solution for resource-constrained environments.

**Index Terms**—Federated learning, over-the-air computation, slimmable neural network, Lyapunov optimization, multi-armed bandit.

## I. INTRODUCTION

WITH the rapid proliferation of the Internet of Things (IoT) and edge intelligence, massive data are naturally generated and stored across large numbers of IoT devices (devices), making traditional centralized learning frameworks

increasingly prone to privacy leakage and communication bottlenecks [1]. Federated learning (FL) has emerged as a promising privacy-preserving distributed training paradigm that allows devices to perform local model updates while transmitting only model parameters or gradients to an edge server (ES) [2]. Despite these advantages, deploying FL over resource-constrained devices remains challenging due to two fundamental limitations: (i) *limited communication resources*, since frequent uplink and downlink exchanges of model parameters over many training rounds can incur substantial communication overhead under constrained bandwidth [3]; and (ii) *heterogeneous client capabilities in computation and energy*, which cause imbalanced local execution times and ultimately hinder global training efficiency [4]. Together, these factors significantly restrict the scalability and performance of FL in large-scale wireless edge networks.

To alleviate the *limited communication resource* bottleneck, over-the-air computation (AirComp) has recently gained considerable attention for exploiting the waveform superposition property of wireless multiple-access channels to enable simultaneous transmission and aggregation of local model updates [5]. When the aggregation noise in AirComp is effectively suppressed, it can substantially reduce communication latency and bandwidth consumption while maintaining satisfactory convergence and model accuracy [6]. In parallel, various model compression techniques, such as quantization [7], [8], sparsification [9], [10], and pruning [11], [12], have been incorporated into FL to further reduce communication and computation overheads. Specifically, quantization and sparsification compress transmitted model updates to mitigate communication costs [13], while pruning eliminates redundant parameters to generate lightweight models, effectively decreasing local computation latency and energy consumption [14].

Nevertheless, integrating model compression with AirComp in FL to jointly address the challenges of *limited communication resources* and *device heterogeneity in computation and energy resources* remains nontrivial. Quantization and sparsification primarily reduce the transmission volume but do not alleviate the local computation load, as the local model structure and iteration count remain unchanged. In contrast, pruning-based methods produce non-identical model architectures across devices, violating the signal alignment requirement of AirComp and causing aggregation misalignment. Although enforcing a uniform global pruning pattern across all devices can preserve alignment [15], it sacrifices adaptability

This paper was supported in part by the Postgraduate Research and Practice Innovation Program of Jiangsu Province under Grant KYCX25\_2894, in part by the Fundamental Research Funds for the Central Universities under Grant 2025-00063, in part by the Graduate Innovation Program of China University of Mining and Technology under Grant 2025WLKXJ208, in part by the High-Tech District of Suzhou City under Grant RC2025001, in part by the Gusu Innovation Project under Grant ZXL2024360, and in part by the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China under Grant JYB2025XDXM118.

Yue Zhang and Guopeng Zhang are with the School of Computer Science and Technology / School of Artificial Intelligence, China University of Mining and Technology, Xuzhou 221116, China (e-mail: yuezhang@cumt.edu.cn; gpzhang@cumt.edu.cn).

Kezhi Wang is with the Department of Computer Science, Brunel University London, Middlesex UB8 3PH, U.K. (e-mail: kezhi.wang@brunel.ac.uk).

Kun Yang is with the State Key Laboratory of Novel Software Technology, Nanjing University, Nanjing 210008, China, Institute of Intelligent Networks and Communications (NINE) and School of Intelligent Software and Engineering, Nanjing University (Suzhou Campus), Suzhou 215163, China, and School of Computer and Information Science, Qinghai Institute of Technology, Xining 810016, China (e-mail: kunyang@nju.edu.cn).

to device heterogeneity, thereby resulting in suboptimal training efficiency.

Recently, the concept of slimmable neural networks (SNNs) [16] has been introduced, allowing a full model to be vertically decomposed into multiple sub-models with varying widths (e.g., by adjusting channels or neurons). Each sub-model can be activated according to device capabilities, where a model of larger width encompasses all narrower sub-models. Compared with conventional model compression, FL based on SNNs (i.e., FedSNN) enables heterogeneous devices to train sub-models with different widths while preserving a shared nested structure, thus providing strong compatibility with AirComp-based aggregation.

Motivated by these observations, this paper proposes a novel framework, termed *Air-FedSNN*, which enables width-heterogeneous SNN training over AirComp-based wireless aggregation to jointly enhance communication, computation, and energy efficiency in resource-heterogeneous IoT networks. To fully realize the potential of *Air-FedSNN*, several key challenges must be addressed:

- *Unknown impact of model widths and aggregation errors:* Devices train local models with different widths and perform AirComp-based aggregation. The varying representational capacities and aggregation errors jointly influence the global model performance. Quantitatively characterizing these coupled effects remains a complex problem for ensuring reliable training.
- *Dynamic width allocation without prior knowledge:* In resource-heterogeneous environments, both device capabilities and channel conditions vary dynamically across communication rounds. Determining model widths that balance resource utilization and model accuracy, without prior knowledge of device states, is a significant challenge.

To address these challenges, a *block-wise model sharing mechanism* is first introduced to ensure parameter alignment among heterogeneous-width local models. Then, a *width-aware aggregation mechanism* is developed to improve global aggregation by adaptively weighting structurally aligned local updates according to the representational capacity of each sub-model. Subsequently, a theoretical convergence analysis is conducted to illustrate the influence of local model width and AirComp-induced errors on global learning performance, thereby revealing how model width affects both aggregation accuracy and convergence behavior. Furthermore, a *dynamic width allocation algorithm* is proposed based on Lyapunov optimization and the multi-armed bandit (MAB) framework, which enables adaptive width selection under long-term energy constraints without requiring prior knowledge of device capabilities. The Lyapunov framework decomposes the long-term stochastic optimization into tractable per-round subproblems, while the MAB strategy adaptively selects model widths to balance energy efficiency and learning performance. The main contributions of this work are summarized as follows:

- 1) A novel framework, *Air-FedSNN*, is proposed, where AirComp and SNN techniques are jointly optimized to mitigate communication, computation, and energy burdens in heterogeneous IoT networks, achieving an effective balance

between model accuracy and training efficiency.

- 2) A block-wise model sharing and width-aware AirComp aggregation mechanism is designed to enhance multi-width aggregation performance. Theoretical analysis is provided to characterize the coupled effects of model width diversity and AirComp-induced aggregation errors on convergence, ensuring stable and reliable training under adaptive-width scenarios.
- 3) An energy-aware dynamic width allocation strategy is developed based on Lyapunov optimization and MAB theory, which adaptively maximizes model width utilization under long-term energy constraints.
- 4) Extensive experiments on MNIST, CIFAR-10 and CIFAR-100 datasets demonstrate that *Air-FedSNN* achieves superior accuracy-energy tradeoffs compared with conventional compression-based FL approaches, validating its effectiveness in resource-heterogeneous environments.

The paper is organized as follows: Section II reviews related works, Section III presents the system framework, Section IV details the convergence analysis, Section V proposes the dynamic width allocation algorithm, Section VI discusses the simulation results, Section VII concludes the paper.

## II. RELATED WORKS

The deployment of FL in wireless edge networks faces critical challenges, primarily stemming from limited communication bandwidth and the heterogeneity of device resources, such as varying computational capacities and energy budgets. To mitigate these bottlenecks and enhance overall learning efficiency, extensive research efforts have been dedicated to optimizing model training and aggregation mechanism. In this section, we review the SNN and AirComp-based FL methods.

1) *AirComp-based FL:* AirComp leverages the superposition property of analog signals to perform model aggregation, thereby improving communication and aggregation efficiency [17]. In [18], a multi-resource allocation mechanism has been proposed to improve the overall learning efficiency of AirComp based FL (*Air-FL*). To balance energy budget with training performance, a dynamic device scheduling method based on Lyapunov optimization is introduced in [19], and training frameworks based on energy harvesting have been explored [20]. To further improve training efficiency, model compression techniques are applied. In [9], analog aggregation based on gradient sparsification and digital transmission based on gradient quantization have been investigated, respectively. Meanwhile, an aggregation mechanism based on 1-bit quantization has been proposed in [7], and the quantitative relationship between model compression and learning performance has been derived by combining sparsification with 1-bit quantization [8]. Moreover, an aggregation method based on global sparsity patterns has been developed to maintain model structural consistency [10]. In [14], the convergence performance of pruning-based *Air-FL* is analyzed, and the importance of global model parameters is evaluated using a loss importance measure. Furthermore, an *Air-FL* method combining pruning and quantization is proposed in [12], where pruning is applied during local training to enhance computa-

tional efficiency, while quantization is employed during model uploading to reduce the transmission data size.

2) *SNN-based FL*: SNN represents a structural-level compression mechanism that does not directly prune weights but instead introduces network scalability. The key idea is to dynamically adjust the network width to create sub-networks of different sizes, enabling devices with heterogeneous computational capabilities to train width-adaptive local models [16]. Recent research has explored the integration of SNNs into FL to accommodate devices with varying resource capabilities [21]–[23]. To address communication and computation heterogeneity across devices, SNNs are first introduced into FL in [21], where superposition coding enables models with different widths to be transmitted according to channel conditions, and a superposition training strategy mitigates interference among local sub-networks. In [22], a scalable active caching framework has been developed by incorporating a federated reinforcement learning algorithm based on SNNs, achieving efficient resource allocation across varying numbers of devices. Furthermore, to tackle both statistical and system heterogeneity, a series of SNN-based FL algorithms has been designed in [23], including device clustering, width allocation, and model aggregation.

3) *Key Differences from Existing Works*: Existing Air-FL and FedSNN methods address communication efficiency and resource heterogeneity from different perspectives. However, their integration in resource-heterogeneous networks, as well as the resulting coupling among wireless aggregation errors, heterogeneous-width local training, and dynamic width allocation, remains underexplored. Different from Air-FL approaches, Air-FedSNN exploits the shared nested structure of SNNs to preserve structural consistency across different sub-models, while allowing devices to adapt their model widths according to heterogeneous resource conditions. Meanwhile, unlike FedSNN methods that mainly rely on conventional digital aggregation, the proposed framework integrates AirComp to improve aggregation efficiency, while explicitly considering aggregation errors, wireless channel conditions, and their coupling with width-varying local training.

Therefore, Air-FedSNN is not a simple combination of AirComp and SNN-based FL. Instead, it provides a communication-efficient and resource-adaptive framework that jointly addresses heterogeneous-width local training, block-wise AirComp aggregation, and dynamic width allocation under unknown device states.

### III. PROPOSED AIR-FEDSNN FRAMEWORK

In this section, we present the proposed Air-FedSNN framework for heterogeneous FL. We first introduce the necessary background and then describe the framework design.

#### A. Preliminary

We first briefly review the FL model and the scalable neural network architecture adopted in this work, which serve as the foundation for the proposed Air-FedSNN framework.

1) *FL Model*: In a wireless edge network, an edge server (ES) coordinates the FL process with a set  $\mathcal{K}$  of  $K$  IoT devices. For each device  $k$  ( $\forall k \in \mathcal{K}$ ), its locally collected dataset is denoted as  $\mathcal{D}_k = \{(\mathbf{x}_{k,i}, y_{k,i})\}_{i=1}^{D_k}$ , where  $D_k = |\mathcal{D}_k|$  represents the dataset size,  $\mathbf{x}_{k,i} \in \mathbb{R}^n$  denotes the  $i$ -th sample, and  $y_{k,i} \in \mathbb{R}$  is the corresponding label. Let  $\mathcal{D} = \bigcup_{k \in \mathcal{K}} \mathcal{D}_k$  denote the aggregated global dataset with total size  $D = \sum_{k=1}^K D_k$ . The objective of FL is to obtain a global parameter vector  $\mathbf{w} \in \mathbb{R}^q$  by minimizing the global loss function [24]:

$$\min_{\mathbf{w}} F(\mathbf{w}) = \sum_{k \in \mathcal{K}} \frac{D_k}{D} F_k(\mathbf{w}), \quad (1)$$

where  $F_k(\mathbf{w})$  denotes the local loss function of device  $k$ .

2) *Slimmable neural network*: IoT devices typically exhibit significant heterogeneity in computational and energy resources, which can considerably affect model training efficiency. To mitigate this limitation, SNNs have been proposed. An SNN maintains a full-width neural network and a candidate width set  $\mathcal{M} = \{\delta_1, \delta_2, \dots, \delta_M\}$ , where  $\delta_m$  denotes the width ratio of the  $m$ -th sub-network relative to the full-width model. By selecting a specific width  $\delta_m$ , only the corresponding proportion of channels or neurons at each layer is activated, thereby switching the full model to a sub-model with width  $\delta_m$ . A larger-width sub-model contains the parameters of all smaller sub-models and additionally activates more channels or neurons. Therefore, SNNs allow devices with different computation and energy capabilities to train sub-models of different widths while maintaining a consistent parameter structure within the same full-width model.

3) *Integrating FL with SNN*: By integrating SNN into FL, i.e., FedSNN, each device can dynamically select a model width according to its available resources and participate in FL training. A natural concern is that gradients from different-width sub-models correspond to different local objectives, which might appear inconsistent with the standard FL assumption that all updates optimize the same global model. However, all sub-models share the same input and output spaces and target the same learning task [16]. Narrower sub-models optimize a masked version of the full-width objective, activating only a subset of parameters. Therefore, these local objectives can be regarded as restrictions of a single global objective rather than independent functions. Moreover, updates are mapped to the full model and aggregated across devices, ensuring consistent updates toward the shared objective. This principle has also been validated in model-heterogeneous FL, where devices with different parameter sizes collaboratively train a shared global model through parameter-aligned aggregation [25], [26].

Although FedSNN alleviates computational heterogeneity, the frequent uplink transmission of high-dimensional model parameters may still lead to substantial communication bottlenecks. To further enhance communication efficiency, AirComp is employed to enable one-shot aggregation of width-varying local models. By integrating AirComp with FedSNN, a unified and communication-efficient framework termed Air-FedSNN is developed. The workflow of Air-FedSNN is shown in Fig. 1 and described as follows.

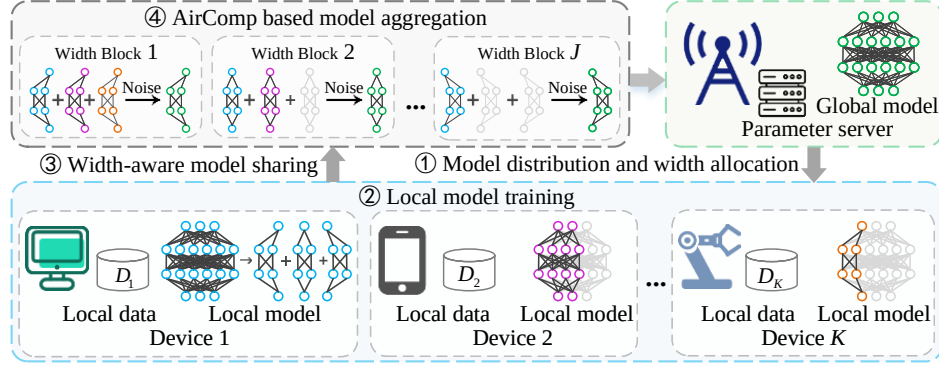


Fig. 1. System model and workflow of the proposed Air-FedSNN framework.

### B. Model Distribution and Width Allocation

Consider an FL process consisting of  $T$  global communication rounds indexed by  $\mathcal{T} = \{1, 2, \dots, T\}$ . In each round  $t \in \mathcal{T}$ , the ES maintains a global model  $\mathbf{w}^t$ . Following the SNN paradigm,  $\mathbf{w}^t$  is divided into  $M$  distinct network widths, where width  $m \in \{1, \dots, M\}$  corresponds to a sub-model with size  $\delta_m$ .

After receiving the initialized global model  $\mathbf{w}^t$  broadcast by the ES, each device  $k$  constructs its local model  $\mathbf{w}_k^t$ . In round  $t$ , device  $k$  performs  $\Gamma$  local iterations, and the local model at iteration  $l$  is denoted by  $\mathbf{w}_k^{t,l}$  ( $l = 1, 2, \dots, \Gamma$ ). The computational and energy resources of each device are assumed constant within each round but may vary across rounds. Therefore, at the beginning of round  $t$ , device  $k$  is assigned an appropriate model width  $\beta_k^t = \delta_m$ ,  $\forall m \in \{1, 2, \dots, M\}$  to match its resource state. Only the sub-model corresponding to width  $\beta_k^t$  is activated, and parameters outside this width are masked. The initialized local model is thus given by

$$\mathbf{w}_k^{t,l=0} = \mathbf{w}^t \odot \Phi_k^t, \quad (2)$$

where  $\Phi_k^t$  is a binary mask matrix indicating the active parameter subset of device  $k$  in round  $t$ .

### C. Local Model Training

At the  $l$ -th local iteration ( $l = 1, 2, \dots, \Gamma$ ), the gradient of the local loss function is computed as

$$\nabla F_k(\mathbf{w}_k^{t,l}) = \frac{1}{|D_k^{t,l}|} \sum_{(\mathbf{x}_k, y_k) \in D_k^{t,l}} \nabla f(\mathbf{w}_k^{t,l}, (\mathbf{x}_k, y_k)), \quad (3)$$

where  $f(\mathbf{w}, (\mathbf{x}_{k,i}, y_{k,i}))$  is the empirical sample-wise loss, and  $D_k^{t,l}$  denotes the mini-batch used in iteration  $l$ .

During local training, *only parameters within the allocated width participate in forward and backward propagation*, while masked parameters are fixed. Using stochastic gradient descent (e.g., mini-batch SGD), the local model is updated as

$$\mathbf{w}_k^{t,l} = \mathbf{w}_k^{t,l-1} - \eta \nabla F_k(\mathbf{w}_k^{t,l-1}), \quad (4)$$

where  $\eta$  denotes the learning rate.

### D. Width-Aware Model Sharing

Owing to the variation in model widths across devices and rounds in Air-FedSNN, standard aggregation strategies such as FedAvg are not directly applicable, as they assume a homogeneous model structure. To enable scalable and consistent aggregation over heterogeneous-width local models, a width-aware model sharing mechanism is introduced:

1) *Width Block Representation*: The full local model  $\mathbf{w}_k^t$  of device  $k$  is partitioned into  $J$  width blocks, where  $1 \leq J \leq M$  and  $M$  denotes the total number of network widths, each corresponding to a fraction  $\frac{1}{M}$  of the overall model width. Block  $j$  represents the incremental parameter region introduced when expanding the model width from  $\delta_{j-1}$  to  $\delta_j$ . The index of each width block is denoted by  $j \in \{1, 2, \dots, J\}$ . In practice,  $J$  is typically set equal to  $M$ , ensuring that each block aligns with a specific width increment. The updated local model  $\mathbf{w}_k^{t+1}$  consists of a subset of these blocks, and gradient aggregation at the ES is executed independently for each block.

Let  $a_{k,j}^t \in \{0, 1\}$  denote a binary variable indicating whether device  $k$  activates block  $j$  in round  $t$ . The set of devices selecting block  $j$  is defined as  $\mathcal{S}_j^t \subseteq \mathcal{K}$ , with size  $|\mathcal{S}_j^t| = |\mathcal{S}_j^t| = \sum_{k \in \mathcal{K}} a_{k,j}^t$ . The selected width of device  $k$  in round  $t$  follows

$$\beta_k^t = \delta_m = \sum_{j=1}^J a_{k,j}^t \frac{1}{M}. \quad (5)$$

2) *Monotonic Width Scheduling*: To maintain structural coherence among heterogeneous-width models, the width activation pattern for any device  $k$  is required to satisfy the monotonicity constraint

$$a_{k,J}^t \leq a_{k,J-1}^t \leq \dots \leq a_{k,1}^t, \quad (6)$$

ensuring that activation of a wider width block implies activation of all narrower blocks.

Consequently, the local model of device  $k$  can be expressed as the concatenation of its activated width components:

$$\mathbf{w}_k^{t+1} = \text{concat} \left( a_{k,1}^t \mathbf{w}_{k,1}^{t+1}, a_{k,2}^t \mathbf{w}_{k,2}^{t+1}, \dots, a_{k,J}^t \mathbf{w}_{k,J}^{t+1} \right), \quad (7)$$

where  $\mathbf{w}_{k,j}^t$  is the local sub-model associated with block  $j$ .

The model parameters corresponding to width block  $j$  are obtained through masking as

$$\mathbf{w}_{k,j}^{t+1} = \begin{cases} \mathbf{w}_k^{t+1} \odot (\Phi_m - \Phi_{m-1}), & \text{if } m > 1, \\ \mathbf{w}_k^{t+1} \odot \Phi_m, & \text{otherwise,} \end{cases} \quad (8)$$

where  $\Phi_m$  is the binary mask matrix for width  $m$ .

### E. AirComp-Based Model Aggregation

Upon completing local training, AirComp is employed to efficiently aggregate the heterogeneous-width local models. For each width block, only devices that activate the corresponding block participate in its aggregation. The aggregated block updates are then mapped back to their corresponding positions in the full-width global model, thereby avoiding direct aggregation between updates with different parameter structures. Specifically, each sub-model block is first amplitude-modulated into analog symbols and transmitted via time-division multiplexing (TDM) or frequency-division multiplexing (FDM) to the ES, ensuring block-level orthogonality throughout the aggregation stage.

The complex channel coefficient between device  $k$  and the ES on width block  $j$  is denoted by  $h_{k,j}^t \in \mathbb{C}$ . To enable reliable AirComp aggregation under heterogeneous channel conditions, every local sub-model  $\mathbf{w}_{k,j}^t$  is normalized into zero-mean, unit-variance symbols  $\mathbf{s}_{k,j}^t$ . The received signal on block  $j$  at the ES is thus given by

$$\mathbf{y}_j^t = \sum_{k \in \mathcal{K}} h_{k,j}^t \mathbf{x}_{k,j}^t + \mathbf{z}_j^t, \quad \forall j \in \mathcal{J}, \quad (9)$$

where  $\mathbf{x}_{k,j}^t = p_{k,j}^t \mathbf{s}_{k,j}^t$  is the transmitted waveform and  $\mathbf{z}_j^t \sim \mathcal{CN}(0, \sigma_z^2 \mathbf{I})$  denotes the AWGN vector.

1) *Width-Aware AirComp Weighting*: The proposed width-aware aggregation is not intended to resolve parameter incompatibility, which is addressed by the block-wise model sharing mechanism. Instead, it adjusts the relative contributions of structurally aligned updates within each width block. Since different devices participate with different effective widths, the influence of each local update on the global aggregation must reflect its selected width. A larger width activates more channels and parameters, thus generally offering higher representation capacity and performance upper bound under the same training setting. This aligns with common observations in prior SNN-based studies that larger widths yield better accuracy at the cost of higher computation [16], [23]. Accordingly, wider sub-models are assigned larger aggregation weights  $e^{\beta_k^t}$  to capture their greater capacity contribution. To achieve the width-aware weighting, each device applies a transmit coefficient

$$p_{k,j}^t = \sigma_j^t a_{k,j}^t e^{\beta_k^t} (h_{k,j}^t)^H / |h_{k,j}^t|^2, \quad (10)$$

where  $\sigma_j^t$  is a power-scaling factor. Since  $\beta \in (0, 1]$ , the resulting weight difference remains bounded, with the largest weight ratio given by  $e^{1.0} / e^{0.25} \approx 2.12$ . Therefore, larger-width models receive only a moderate preference during aggregation, rather than dominating the global aggregation.

Substituting this expression into the received signal yields

$$\mathbf{y}_j^t = \sum_{k \in \mathcal{K}} \sigma_j^t a_{k,j}^t e^{\beta_k^t} \mathbf{s}_{k,j}^t + \mathbf{z}_j^t, \quad \forall j \in \mathcal{J}, \quad (11)$$

with communication power consumption

$$P_{k,j}^{t,\text{com}} = \mathbb{E} \left[ |p_{k,j}^t \mathbf{s}_{k,j}^t|^2 \right] = \frac{(a_{k,j}^t \sigma_j^t e^{\beta_k^t})^2}{\|h_{k,j}^t\|^2}. \quad (12)$$

2) *AirComp Reconstruction of Width Blocks*: After receiver-side scaling and de-normalization at the ES, the aggregated sub-model on each block  $j$  is reconstructed as

$$\mathbf{w}_j^{t+1} = \sum_{k \in \mathcal{K}} \rho_{k,j}^t \mathbf{w}_{k,j}^{t+1} + \tilde{\mathbf{z}}_j^t, \quad \forall j \in \mathcal{J}, \quad (13)$$

where

$$\rho_{k,j}^t = \frac{a_{k,j}^t e^{\beta_k^t}}{\sum_{i=1}^K a_{i,j}^t e^{\beta_i^t}}, \quad \tilde{\mathbf{z}}_j^t = \frac{\mathbf{z}_j^t}{\sigma_j^t \sum_{i=1}^K a_{i,j}^t e^{\beta_i^t}}. \quad (14)$$

Finally, the global model is reconstructed by concatenating all aggregated width blocks:

$$\mathbf{w}^{t+1} = \text{concat}(\mathbf{w}_1^{t+1}, \mathbf{w}_2^{t+1}, \dots, \mathbf{w}_J^{t+1}). \quad (15)$$

**Remark 1.** The above formulation can be generalized to incorporate both data size and model width as  $\rho_{k,j}^t = a_{k,j}^t D_k e^{\beta_k^t} / \sum_{i \in \mathcal{K}} a_{i,j}^t D_i e^{\beta_i^t}$ . When  $D_k$  is identical across devices, the formulation reduces to the original width-aware weighting. This generalization may increase transmission power requirements, which can be mitigated by appropriately adjusting the block-wise scaling factor  $\sigma_j^t$  for power control.

**Remark 2.** The above AirComp aggregation assumes perfect CSI for channel inversion. In practical wireless systems, however, the transmitter may only obtain an estimated channel, i.e.,  $\hat{h}_{k,j}^t = h_{k,j}^t + \Delta h_{k,j}^t$ , where  $\Delta h_{k,j}^t$  denotes the estimation error. In this case, channel inversion based on  $\hat{h}_{k,j}^t$  introduces a mismatch factor  $\psi_{k,j}^t = h_{k,j}^t (\hat{h}_{k,j}^t)^H / |\hat{h}_{k,j}^t|^2$  into the received signal. Therefore, the aggregated update on width block  $j$  is distorted by the factor  $\psi_{k,j}^t$ , instead of being perfectly aligned as in the ideal case with  $\psi_{k,j}^t = 1$ . This indicates that imperfect CSI affects both the transmit power and the AirComp aggregation error.

Although Air-FedSNN enables scalable training under heterogeneous model widths, the incorporation of width-dependent weighting and AirComp error introduces nontrivial challenges for global convergence and robustness under imperfect wireless channels. These factors necessitate a rigorous characterization of the algorithm's convergence behavior.

## IV. CONVERGENCE ANALYSIS FOR AIR-FEDSNN

In this section, we conduct convergence analysis to quantify the joint impact of multi-width model and AirComp aggregation errors on learning performance, and to establish the corresponding convergence bounds.

First, to facilitate the theoretical analysis, we introduce the following assumptions as in [23], [27] and [28].

**Assumption 1.** (Lipschitz Continuity and Smoothness): For any model parameter  $\mathbf{w}, \mathbf{v} \in \mathbb{R}^q$ , the gradient of loss functions  $\nabla F(\mathbf{w})$  is Lipschitz continuous with a constant  $L > 0$ , i.e.,

$$\|\nabla F(\mathbf{w}) - \nabla F(\mathbf{v})\| \leq L \|\mathbf{w} - \mathbf{v}\|, \quad (16)$$

which can be extended to

$$F(\mathbf{w}) \leq F(\mathbf{v}) + \langle \nabla F(\mathbf{v}), \mathbf{w} - \mathbf{v} \rangle + \frac{L}{2} \|\mathbf{w} - \mathbf{v}\|^2, \quad (17)$$

**Assumption 2.** (Unbiased Stochastic Gradient): The local stochastic gradient at any width block  $j$  is assumed to be unbiased, i.e.,

$$\mathbb{E} \left[ \sum_{k=1}^K \rho_{k,j} \nabla F_{k,j}(\mathbf{w}) \right] = \nabla F_j(\mathbf{w}). \quad (18)$$

**Assumption 3.** (Model Width Bound): The change of global model version within total training period  $T$  is bounded by a positive constant  $\varphi$ , i.e.,

$$\|\mathbf{w} - \mathbf{w} \odot \Phi_k\|^2 \leq \varphi^2 \|W\|^2 \quad (19)$$

where  $\varphi \in [0, 1]$  represents a uniform upper bound on the relative discrepancy induced by width masking, and  $\|W\|^2$  denotes an upper bound on the squared norm of the full-width model.

**Assumption 4.** (Gradient Bound): The expected squared norm of gradient  $\mathbb{E} [\|\nabla F_k(\mathbf{w})\|^2]$  at any device  $k$  is bounded by a positive constant  $G$ , i.e.,

$$\mathbb{E} [\|\nabla F_k(\mathbf{w})\|^2] \leq G^2. \quad (20)$$

Based on **Assumption 3** and **Assumption 4**, we first present the following lemmas for the convergence analysis.

**Lemma 1.** The expected squared norm of the discrepancy between the locally updated model  $\mathbf{w}_k^{t,l}$  and the global model  $\mathbf{w}^t$  from the previous communication round is bounded by

$$\mathbb{E} [\|\mathbf{w}_k^{t,l} - \mathbf{w}^t\|^2] \leq 2\eta^2 \Gamma^2 G^2 + 2\varphi^2 \|W\|^2 \quad (21)$$

*Proof.* This proof is derived from Assumptions 3 and 4, and is omitted due to space limitations.  $\square$

Next, the convergence properties of Air-FedSNN are derived to characterize the joint influence of width allocation strategies and AirComp-induced aggregation errors on the evolution of the global model. Since practical learning objectives in modern neural networks are typically non-convex, the analysis is conducted under the non-convex setting.

In non-convex optimization, the expected squared gradient norm serves as a standard metric for assessing convergence [29], [30]. Accordingly, the global loss function is said to reach a  $\xi$ -suboptimal stationary solution if

$$\frac{1}{T} \sum_{t=1}^T \mathbb{E} [\|\nabla F(\mathbf{w}^t)\|^2] \leq \xi, \quad (22)$$

which ensures that the iterative procedure approaches a stationary point even when global optimality is unattainable.

The following theorem provides sufficient conditions for the convergence of Air-FedSNN. The bound explicitly reflects the combined contributions of network-width heterogeneity, width-aware weighting, and AirComp aggregation errors.

**Theorem 1.** Let  $F(\mathbf{w}^1)$  and  $F^*$  denote the initial and optimal loss values. For a learning rate satisfying  $0 < \eta < \frac{1}{2L\Gamma}$ , after

$T$  communication rounds, the time-averaged expected squared norm of the gradient  $\frac{1}{T} \sum_{t=1}^T \mathbb{E} [\|\nabla F(\mathbf{w}^t)\|^2]$  is bounded as

$$\begin{aligned} \frac{1}{T} \sum_{t=1}^T \mathbb{E} [\|\nabla F(\mathbf{w}^t)\|^2] &\leq \underbrace{\frac{2}{T\eta\Gamma(1-2L\eta\Gamma)} \mathbb{E}[F(\mathbf{w}^1) - F^*]}_A \\ &+ \underbrace{\frac{2eL^2K\eta^2\Gamma^2 + 4eL^3K\eta^3\Gamma^3}{\hat{S}(1-2L\eta\Gamma)} G^2}_B + \underbrace{\frac{2eL^2K + 4eL^3K\eta\Gamma}{\hat{S}(1-2L\eta\Gamma)} \varphi^2 \|W\|^2}_C \\ &+ \underbrace{\frac{L}{T\eta\Gamma(1-2L\eta\Gamma)} \sum_{t=1}^T \sum_{j=1}^J \frac{\sigma_z^2}{(S_j^t \sigma_j^t)^2}}_D, \end{aligned} \quad (23)$$

where  $\hat{S} \leq S_j^t, \forall t, j$  denotes the minimum number of devices participating in width block  $j$  across all communication rounds.

*Proof.* Please refer to Appendix A.  $\square$

The RHS of (23) provides an explicit bound on the convergence error. Term  $A$  reflects the initial optimality gap and diminishes with larger  $T$ . Term  $B$  corresponds to the boundedness of the local gradients influenced by width differences. Term  $C$  captures the mismatch between the full-width global model and locally activated sub-models, which represents the parameter-space discrepancy induced by width masking rather than incompatible parameter coordinates across different model architectures. The norm term  $\varphi^2 \|W\|^2$  provides an upper bound on the width-induced discrepancy. Term  $D$  quantifies the AirComp aggregation distortion accumulated across width blocks. The convergence bound in Theorem 1 leads to the following insights:

- 1) Increasing the number of communication rounds  $T$  reduces the initial optimality gap in term  $A$ , thereby accelerating convergence.
- 2) Larger discrepancies between the global and the local model widths increase term  $C$ , thereby enlarging the overall convergence bound. Hence, balanced width allocation is essential for stable training.
- 3) The set of participating widths is determined by the assigned model widths of devices, where a device with a larger width participates in all lower-width blocks, so that each width block consists of devices whose allocated widths cover that block and defines the participation number  $S_j^t$  in each block.
- 4) The participating devices in each width block determine the block-level participation number  $S_j^t$ , which influences terms  $B$ ,  $C$ , and  $D$ . Increasing  $S_j^t$  reduces AirComp-induced errors, improves aggregation accuracy, and promotes faster convergence. When only a few devices support large widths, high-width blocks are updated by fewer devices, resulting in increased AirComp aggregation error and a looser convergence bound.
- 5) The allocated model widths determine the set of participating devices in each width block, thereby affecting  $S_j^t$ . Since the AirComp error term is proportional to

$\sigma_z^2/(S_j^t \sigma_j^t)^2$ , width heterogeneity indirectly influences aggregation accuracy through the participation pattern.

Since different model widths activate distinct sub-networks, each local sub-model optimizes an objective associated with the corresponding part of the full-width global objective. Under non-IID data, the aggregated block gradient may no longer be an unbiased estimate of the corresponding global block gradient. To characterize this discrepancy, we introduce the following assumption.

**Assumption 5. (Gradient Divergence Bound):** Under the non-IID data distribution, the deviation between the local gradient of device  $k$  and the corresponding global gradient at width block  $j$  is bounded by a positive constant  $\zeta^2$ , i.e.,

$$\mathbb{E} \left[ \|\nabla F_{k,j}(\mathbf{w}) - \nabla F_j(\mathbf{w})\|^2 \right] \leq \zeta^2. \quad (24)$$

Based on this assumption, we further derive the convergence bound under the non-IID data distribution.

**Theorem 2.** For the non-IID setting, when the learning rate satisfies  $0 < \eta < \frac{1}{4L\Gamma}$ , the upper bound of the average expected squared gradient after  $T$  communication rounds is given by

$$\begin{aligned} & \frac{1}{T} \sum_{t=1}^T \mathbb{E} \left[ \|\nabla F(\mathbf{w}^t)\|^2 \right] \\ & \leq \frac{2}{T\eta\Gamma(1-4L\eta\Gamma)} \mathbb{E} [F(\mathbf{w}^1) - F^*] + \frac{2(1+2L\eta\Gamma)}{1-4L\eta\Gamma} J\zeta^2 \\ & \quad + \frac{4eL^2K\eta^2\Gamma^2 + 4eL^3K\eta^3\Gamma^3}{\hat{S}(1-4L\eta\Gamma)} G^2 \\ & \quad + \frac{4eL^2K + 4eL^3K\eta\Gamma}{\hat{S}(1-4L\eta\Gamma)} \varphi^2 \|W\|^2 \\ & \quad + \frac{L}{T\eta\Gamma(1-4L\eta\Gamma)} \sum_{t=1}^T \sum_{j=1}^J \frac{\sigma_z^2}{(S_j^t \sigma_j^t)^2}. \end{aligned} \quad (25)$$

*Proof.* The proof extends Theorem 1 to the non-IID case by explicitly bounding the local gradient deviation terms, and is omitted due to space limitations.  $\square$

Compared with the IID case, the non-IID bound contains an additional divergence term  $\frac{2(1+2L\eta\Gamma)}{1-4L\eta\Gamma} J\zeta^2$ . This term captures gradient divergence caused by data heterogeneity and block-dependent device participation. A larger  $\zeta^2$ , i.e., more severe non-IID, leads to a looser convergence bound.

These observations indicate that larger local model widths can improve model convergence. However, hardware constraints at devices necessitate a width allocation policy that balances convergence performance with local computational and energy limitations.

## V. ENERGY-AWARE DYNAMIC WIDTH ALLOCATION

From **Theorem 1**, we observe that increasing the model width in Air-FedSNN effectively mitigates AirComp-induced aggregation errors, thereby improving global convergence performance. However, wider sub-models activate more

neurons and channels, resulting in substantially more connections and trainable parameters. Consequently, increasing the network width directly raises both computation and communication energy consumption at each device. This reveals an inherent *performance–energy tradeoff*: wider models enhance learning accuracy but induce higher local resource demands.

To characterize and optimize this tradeoff, we first quantify the parameter size associated with each width level and derive the corresponding computation and communication energy models. Based on these models, Sec. A formulates the long-term energy-constrained width allocation problem. Given the temporal coupling and stochastic device dynamics, Sec. B transforms the problem into per-round subproblems using Lyapunov optimization. Finally, Sec. C develops an online MAB algorithm for adaptive width allocation under uncertainty.

### A. Problem Formulation

The number of parameters associated with the  $j$ -th width block is

$$\gamma_j = \begin{cases} \|\Phi_m - \Phi_{m-1}\|_0, & m > 1, \\ \|\Phi_m\|_0, & \text{otherwise.} \end{cases} \quad (26)$$

Thus, the total number of active parameters of device  $k$  in round  $t$  is  $b_k^t = \sum_{j=1}^J a_{k,j}^t \gamma_j$ , corresponding to the parameters associated with its assigned width  $\beta_k^t$ .

Let  $c$  denote the number of CPU cycles required to train a full-width model on a single sample, and let  $f_k^t$  denote the instantaneous CPU frequency of device  $k$ . Then the local training energy consumption is modeled as

$$E_k^{t,\text{trn}} = \xi_k (f_k^t)^2 \Gamma D_k \frac{b_k^t c}{q}, \quad (27)$$

where  $b_k^t/q$  denotes the proportion of active parameters and reflects the actual computational load.

For communication, the transmission energy of device  $k$  for sending the  $j$ -th width block is

$$E_{k,j}^{t,\text{com}} = P_{k,j}^t \gamma_j \tau = \frac{(a_{k,j}^t \sigma_j^t e^{\beta_k^t})^2 \gamma_j \tau}{\|h_{k,j}^t\|^2}, \quad (28)$$

where  $\tau$  is the per-parameter transmission duration. The communication and local energy consumption at round  $t$  are

$$E_k^{t,\text{com}} = \sum_{j=1}^J E_{k,j}^{t,\text{com}}, \quad E_k^t = E_k^{t,\text{trn}} + E_k^{t,\text{com}}. \quad (29)$$

From **Theorem 1**, we observe that increasing the model width effectively mitigates AirComp based aggregation error, thereby improving model convergence performance. Meanwhile, as shown in eqs. (27) and (28), the network width directly determines both the computation and communication energy consumption on each device. This reveals an inherent trade-off in which increasing the model width can potentially improve learning performance while simultaneously raising the computation and communication burden.

To further validate this tradeoff, we conduct a preliminary experiment on the CIFAR-10 dataset using a two-layer CNN

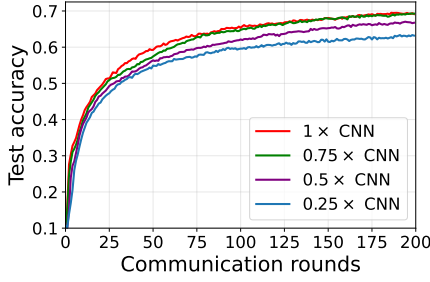


Fig. 2. Test accuracy across different network widths.

TABLE I  
ENERGY PERFORMANCE COMPARISON ON CIFAR-10 DATASET.

| Metrics         | 1×CNN  | 0.75×CNN | 0.5×CNN | 0.25×CNN |
|-----------------|--------|----------|---------|----------|
| Energy          | 3.81   | 2.16     | 0.97    | 0.25     |
| Increase Factor | 15.24× | 8.64×    | 3.88×   | 1×       |

partitioned into four width levels ( $1\times$ ,  $0.75\times$ ,  $0.5\times$ , and  $0.25\times$ ). The experiment involves 20 devices, 200 communication rounds, and a learning rate of 0.01, where each device is assigned a fixed network width. As illustrated in Fig. 2, the Air-FedSNN method with wider sub-models achieves faster convergence and higher accuracy due to stronger representation capacity. However, Table I shows that the accuracy improvements come at a substantial energy cost, with increasing the width from  $0.25\times$  to  $1\times$  leading to over a  $15\times$  rise in energy cost. These empirical findings are consistent with the theoretical insight regarding the tradeoff between resource efficiency and learning performance.

Motivated by these findings, we adopt model width as a tractable surrogate for learning performance. Accordingly, we design a dynamic width-allocation mechanism that assigns the wider feasible sub-model to each device while respecting per-device long-term energy constraints. This leads to the following optimization problem:

$$\mathcal{P}0: \max \sum_{t=1}^T \sum_{k=1}^K \beta_k^t, \quad (30a)$$

$$\text{s.t.} \quad \frac{1}{T} \sum_{t=1}^T E_k^t \leq \bar{E}_k, \quad \forall k \in \mathcal{K}, \quad (30b)$$

$$\beta_k^t \in (0, 1], \quad \forall k, t. \quad (30c)$$

where  $\beta_k^t$  denotes the network width of device  $k$  in communication round  $t$ , with  $\beta_k^t = 1$  indicating a full-width model. The energy consumption during model training is limited by the budget in constraint (30b). Meanwhile, constraint (30c) defines the set of available network widths for device  $k$  at round  $t$ , which inherently encompasses the monotonicity constraint specified in eq. (6).

Although the optimization variable is the model width, the resulting decisions are tightly coupled with wireless communication resources and AirComp aggregation performance. From an edge network perspective, the proposed framework reveals the following insights.

- 1) *Model width as a communication-resource adaptation variable.* A larger width activates more parameter blocks, thereby increasing the number of transmitted symbols and the corresponding communication energy consumption.

Under heterogeneous energy budgets and channel conditions, width allocation can therefore be viewed as an adaptive communication-resource allocation mechanism rather than merely a model-design choice.

- 2) *Width allocation affects AirComp aggregation quality via participation.* The selected width determines the activated parameter blocks and thus the block-level participation number  $S_j^t$ . Since the aggregation distortion decreases with  $(S_j^t)^2$ , width allocation influences the AirComp aggregation quality through the coupling  $\beta_k^t \rightarrow S_j^t \rightarrow \text{aggregation error}$ . Therefore, model width influences not only communication cost but also OTA aggregation reliability and learning performance.
- 3) *Wireless conditions determine learning performance.* The transmission power, channel gain, and wireless noise jointly affect both communication energy consumption and aggregation error. Therefore, the optimal width allocation is inherently communication-aware and depends on the underlying wireless environment.

Problem  $\mathcal{P}0$  is challenging due to: 1) *Temporal coupling due to energy constraints:* The long-term energy budget constraint in eq. (30b) introduces temporal dependencies across communication rounds. Decisions made in any given round  $t$  directly influence the width allocation in subsequent rounds, thereby complicating the optimization process. 2) *Dynamic device resources and lack of prior knowledge:* The resource status of devices (e.g., CPU cycle frequency and channel conditions), fluctuates over time. The system lacks prior knowledge of the devices' capabilities, making it difficult to optimally allocate resources based on real-time conditions.

## B. Problem Transformation via Lyapunov Optimization

To handle the long-term constraint in  $\mathcal{P}0$ , we adopt Lyapunov optimization to convert the problem into per-round decisions [31]. First, we introduce a virtual energy queue  $U_k^t$  for device  $k$ , which is updated according to its energy consumption in each round:

$$U_k^{t+1} = \max(U_k^t + E_k^t - \bar{E}_k, 0). \quad (31)$$

The queue tracks the accumulated energy cost over time, ensuring that the device's energy usage does not exceed the predefined budget.

Then, to indicate the queue status, we define the Lyapunov function as the sum of the squared values of the virtual energy queues across all devices, i.e.,

$$L(\Theta^t) = \frac{1}{2} \sum_{k=1}^K (U_k^t)^2, \quad (32)$$

where  $\Theta^t = \{U_k^t, \forall k \in \mathcal{K}\}$  represent the set of all queue backlogs at time  $t$ . The goal is to minimize the drift of the Lyapunov function, which ensures the energy constraints. The Lyapunov drift is represented as

$$\Delta(\Theta^t) = \mathbb{E} [L(\Theta^{t+1}) - L(\Theta^t) | \Theta^t]. \quad (33)$$

To stabilize the queues while optimizing the system's performance, we adopt the *drift-plus-penalty* method, which adds a penalty term for violating the energy constraints:

$$\Delta_V(\Theta^t) = \Delta(\Theta^t) - V\mathbb{E}\left[\sum_{k=1}^K \beta_k^t \mid \Theta^t\right], \quad (34)$$

where  $V \geq 0$  is a control parameter balancing cost minimization and queue stability. The penalty term is designed to penalize the excess energy consumption of devices. Given the virtual queue state  $\Theta^t$ , the *drift-plus-penalty* can be upper-bounded as

$$\begin{aligned} \Delta_V(\Theta^t) &= \Delta(\Theta^t) - V\mathbb{E}\left[\sum_{k=1}^K \beta_k^t \mid \Theta^t\right], \\ &\leq \frac{1}{2} \sum_{k=1}^K \left[ (E_k^t)^2 + (\bar{E}_k)^2 - 2U_k^t \bar{E}_k \right] \\ &\quad + \sum_{k=1}^K U_k^t \mathbb{E}[E_k^t \mid \Theta^t] - V\mathbb{E}\left[\sum_{k=1}^K \beta_k^t \mid \Theta^t\right] \end{aligned} \quad (35)$$

Based on the theory of Lyapunov optimization, minimizing the derived upper bound in (35) by discarding the constant terms is equivalent to solving the following maximization problem for each communication round  $t$ :

$$\mathcal{P}1: \max_{\{\beta_k^t\}} V \sum_{k=1}^K \beta_k^t - \sum_{k=1}^K U_k^t E_k^t, \quad (36a)$$

$$\text{s.t. } \beta_k^t \in (0, 1], \quad \forall k \in \mathcal{K}. \quad (36b)$$

To avoid wide-model dominance, resource-limited devices still contribute to the shared core blocks activated by all widths, ensuring their participation in the global aggregation. Meanwhile, the Lyapunov energy queue penalizes high energy cost, thereby discouraging continuous allocation of large-width models and promoting balanced width usage across devices.

Although  $\mathcal{P}1$  can be solved per round, the lack of prior knowledge of device states prevents direct optimization, motivating an online learning approach.

### C. Online Width Allocation via Multi-Armed Bandits

This subsection presents an online width allocation strategy based on the MAB framework with Upper Confidence Bound (UCB) estimation [32]. The objective is to enable each device to dynamically choose an appropriate network width while accounting for its energy state and historical performance. In the proposed formulation, each device is modeled as an independent decision maker, where the candidate width levels correspond to different arms in the MAB setting. At each communication round, the device selects an arm to balance the tradeoff between exploiting historically favorable width levels and exploring alternative configurations that may yield improved performance under time-varying system conditions.

To capture temporal variations in the environment, we adopt a discounted learning mechanism. The discounted number of selections for arm  $m$  by device  $k$  up to round  $t$  is defined as

$$N_{k,m}^t(\lambda) = \sum_{l=1}^t \lambda^{t-l} \mathbb{I}\{\beta_k^l = \beta_m\}, \quad \forall m \in \mathcal{M}, \quad (37)$$

where  $0 \leq \lambda \leq 1$  is the discount factor, enabling the system to emphasize recent observations. The normalized temporal weight is given by

$$n^t(\lambda) = \sum_{l=1}^t \lambda^{t-l}. \quad (38)$$

To evaluate the historical utility of each width choice, we maintain discounted empirical accumulations of width selections and energy consumptions:

$$\Phi_{k,m}^t(\lambda) = \sum_{l=1}^t \lambda^{t-l} \beta_m \mathbb{I}\{\beta_k^l = \beta_m\}, \quad (39)$$

$$\Upsilon_{k,m}^t(\lambda) = \sum_{l=1}^t \lambda^{t-l} E_{k,m}^l \mathbb{I}\{\beta_k^l = \beta_m\}. \quad (40)$$

The corresponding discounted empirical averages are computed as

$$\bar{\Phi}_{k,m}^t(\lambda) = \frac{\Phi_{k,m}^t(\lambda)}{N_{k,m}^t(\lambda)} + \sqrt{\frac{2\kappa \log(n^t(\lambda))}{N_{k,m}^t(\lambda)}}, \quad (41)$$

$$\bar{\Upsilon}_{k,m}^t(\lambda) = \frac{\Upsilon_{k,m}^t(\lambda)}{N_{k,m}^t(\lambda)} + \sqrt{\frac{2\kappa \log(n^t(\lambda))}{N_{k,m}^t(\lambda)}}, \quad (42)$$

where the exploration term  $\sqrt{\frac{2\kappa \log(n^t(\lambda))}{N_{k,m}^t(\lambda)}}$  is introduced to account for the uncertainty arising from limited sample sizes, with  $\kappa$  controlling the intensity of exploration.

The UCB metric that synthesizes both learning performance and energy cost for arm  $m$  of device  $k$  is defined as

$$\text{UCB}_{k,m}^t(\lambda) = V \cdot \bar{\Phi}_{k,m}^t(\lambda) - U_k^t \cdot \bar{\Upsilon}_{k,m}^t(\lambda), \quad (43)$$

where  $V$  is the drift-plus-penalty control parameter and  $U_k^t$  denotes the Lyapunov queue reflecting the device's energy deviation. The next-round width selection follows

$$\beta_k^{t+1} = \frac{1}{M} \cdot \arg \max_{m \in \mathcal{M}} \text{UCB}_{k,m}^t(\lambda). \quad (44)$$

This decision rule allows the device to favor width levels offering higher utility while penalizing excessive energy usage, thereby remaining aligned with the long-term constraint.

### D. Overall Algorithm

The proposed online width allocation method is detailed in **Algorithm 1**. At each round  $t$ , lines 3–6 implement the core decision-making stage at the ES. Specifically, for every device  $k$ , the ES evaluates the utility of each arm  $m$  by computing its corresponding UCB value according to (43). Based on these UCB scores, the ES assigns the width  $\beta_k^t$  following the selection rule in (44). Once the widths are determined, line 7 triggers the dissemination of both the updated global model and the allocation results to all devices. Lines 8–11 then carry out the decentralized update phase: each device initializes and trains its sub-model using the allocated width (line 8), after which the normalized model updates are transmitted in width-specific blocks  $j \in \mathcal{J}$  (line 10). Line 12 aggregates the received signals  $\mathbf{y}_j^t$  from all blocks and reconstructs the global model  $\mathbf{w}^{t+1}$  according to (15). Finally, lines 13–14 update the

virtual queues  $U_k^{t+1}$  based on the estimated per-device energy expenditure as dictated by (31).

Overall, the iterative steps in **Algorithm 1** jointly coordinate width selection, local computation, and over-the-air aggregation. By systematically coupling exploration, performance estimation, and energy-aware queueing, the algorithm enables adaptive width decisions across heterogeneous devices while preserving the convergence efficiency of Air-FedSNN.

---

**Algorithm 1:** Online width allocation algorithm for Air-FedSNN.

---

```

1 Initialization: Global model  $\mathbf{w}^1$ , local models
    $\mathbf{w}_k^1 = \mathbf{w}^1$ , virtual queues  $U_k^1 = 0$ .
2 for  $t = 1, 2, \dots, T$  do
3   for  $k \in \mathcal{K}$  do
4     Compute  $\text{UCB}_{k,m}^t(\lambda)$  for all  $m$  via (43);
5     Select width  $\beta_k^t$  according to (44);
6   end
7   ES broadcasts  $\mathbf{w}^t$  and the width assignments to all
   devices;
8   for  $k \in \mathcal{K}$  do
9     Update local sub-model using (4);
10    Transmit normalized parameters over different
    width blocks via AirComp;
11  end
12  ES reconstructs  $\mathbf{w}^{t+1}$  from  $\mathbf{y}_j^t$ ;
13  ES estimates device energy consumption;
14  Update queues  $U_k^{t+1}$  using (31);
15 end

```

---

## VI. SIMULATION RESULTS

This section provides a comprehensive evaluation of the proposed Air-FedSNN framework. The experimental setup is as follows.

1) *Datasets and Training Model:* We consider widely used datasets including MNIST, CIFAR-10 and CIFAR-100. MNIST comprises 70,000 grayscale images of handwritten digits ( $28 \times 28$ ), with 60,000 for training and 10,000 for testing. CIFAR-10 includes 60,000 color images ( $32 \times 32$ ) across 10 classes, with 50,000 training and 10,000 testing samples. CIFAR-100 has the same image size and total number of samples as CIFAR-10 but spans 100 fine-grained classes, each with 500 training images and 100 testing images.

The training model is a 7-layer CNN consisting of two  $5 \times 5$  convolutional layers with 32 and 64 channels, respectively, each followed by max pooling. The convolutional feature extractor is followed by two fully connected layers with 512 hidden units and an output layer whose dimension matches the number of classes, i.e., 10 for MNIST and CIFAR-10, and 100 for CIFAR-100. A softmax layer is then applied for classification. To achieve SNN, we divided the global model into 4 widths (i.e.,  $0.25 \times$ ,  $0.5 \times$ ,  $0.75 \times$  and  $1 \times$ ).

2) *Data Distribution:* IID and non-IID distributions are both simulated. For the IID case, each device receives 500 (for CIFAR-10 and CIFAR-100) or 600 (for MNIST) random training samples. In the non-IID case, the training data are

partitioned using the Dirichlet distribution  $\text{Dir}(\alpha)$ , where  $\alpha$  controls the degree of data heterogeneity [33]. In addition, synthetic label-skew non-IID distributions are also considered, parameterized by  $\mu$ . For CIFAR-10, quantity-based label skew is adopted [34], where  $\mu$  denotes the label dominance ratio, i.e., a proportion  $\mu$  of samples on each device belongs to a single dominant class. For CIFAR-100, distribution-based label skew (pathological partitioning) is used, where  $\mu$  represents the available classes, meaning each device contains samples from only a subset of  $\mu \times 100\%$  global classes [35].

3) *System setup:* We consider an ES and  $K = 20$  devices randomly deployed in the FL system, with the devices uniformly distributed within a 50m radius around the ES.

- For communication model, the wireless channel between device  $k$  and the ES is modeled as  $h_k^t = \sqrt{L(d_k)} H_k^t$ , where  $d_k$  denotes the distance from devices to the ES,  $L(d_k) = L_0(d_k/d_0)^{-\alpha_d}$  represents the large-scale path loss with reference distance  $d_0 = 1$  and path-loss exponent  $\alpha_d = 2$ , and  $H_k^t$  represents the small-scale Rician fading. The power scaling factor is set to  $\sigma_j = 0.1$  for all  $j \in \mathcal{J}$ , while the noise variance and transmission duration are  $\sigma^2 = 0.0001$  and  $\tau = 0.0001$ , respectively.
- For computation model, the switch capacitance coefficient is set to  $\xi_k = 2 \times 10^{-29}$ , and the number of CPU cycles required to process one data sample is  $c = 10^6$ . To model computation heterogeneity across devices, the baseline CPU frequency of each device is randomly sampled as  $\bar{f}_k \sim \mathcal{U}(2, 5)$  GHz. To capture temporal variations in available CPU frequency, the per-round CPU frequency is sampled from a Gaussian distribution centered at  $\bar{f}_k$  with a variance of 0.5.

The average energy budget for each device is set as  $\bar{E}_k = \bar{E}, \forall k$ , with  $\bar{E} = 0.4$  J for MNIST and  $\bar{E} = 1.0$  J for CIFAR-10 and CIFAR-100, where the higher budget for CIFAR datasets is due to their larger model size and more local iterations. The learning rate is  $\eta = 0.01$  and the Lyapunov control parameter is set to  $V = 10$ . The total communication rounds are 60 for MNIST and 200 for CIFAR-10 and CIFAR-100, with 1 and 5 local updates per round, respectively.

4) *Performance Metrics:* To assess the performance of Air-FedSNN, we use the following evaluation metrics:

- *Test Accuracy:* The classification accuracy of the global model on test dataset, reflecting the overall learning performance and convergence rate of different FL methods.
- *Energy Consumption:* We evaluate both the average per-device energy consumption per round and the average system-wide energy consumption per round.
- *Normalized Cumulative Energy [19]:* Maximum per-device energy usage normalized by the energy budget:  $\max_k \frac{\sum_{i=1}^T E_k^i}{T \bar{E}_k}$ .

5) *Baselines:* To evaluate the effectiveness of the proposed Air-FedSNN framework, we compare it with the following representative baselines, all implemented under the AirComp aggregation setting:

- **Air-FL:** All devices perform local model training using the full-width model in every round, corresponding to the standard FedAvg method without width reduction.

TABLE II  
PERFORMANCE COMPARISON OF DIFFERENT FL METHODS ACROSS DATASETS.

| FL Methods                | MNIST    |                       | CIFAR-10 |                       | CIFAR-100 |                       |
|---------------------------|----------|-----------------------|----------|-----------------------|-----------|-----------------------|
|                           | Accuracy | Energy                | Accuracy | Energy                | Accuracy  | Energy                |
| Air-FL                    | 97.21    | 1.25 (1 $\times$ )    | 69.52    | 3.82 (1 $\times$ )    | 40.21     | 3.86 (1 $\times$ )    |
| Fixed Width 0.75 $\times$ | 97.21    | 0.71 (0.56 $\times$ ) | 69.32    | 2.16 (0.57 $\times$ ) | 38.03     | 2.23 (0.58 $\times$ ) |
| Fixed Width 0.5 $\times$  | 96.77    | 0.32 (0.25 $\times$ ) | 67.04    | 0.97 (0.25 $\times$ ) | 33.59     | 1.05 (0.27 $\times$ ) |
| Random Width              | 96.93    | 0.59 (0.47 $\times$ ) | 67.02    | 1.71 (0.45 $\times$ ) | 35.48     | 1.86 (0.48 $\times$ ) |
| Global Pruning 25%        | 96.92    | 0.71 (0.56 $\times$ ) | 68.74    | 2.16 (0.57 $\times$ ) | 36.37     | 2.23 (0.58 $\times$ ) |
| Global Pruning 50%        | 96.66    | 0.32 (0.25 $\times$ ) | 64.90    | 0.97 (0.25 $\times$ ) | 30.09     | 1.05 (0.27 $\times$ ) |
| Air-FedSNN (Ours)         | 97.22    | 0.57 (0.45 $\times$ ) | 68.72    | 1.02 (0.26 $\times$ ) | 36.17     | 1.04 (0.27 $\times$ ) |

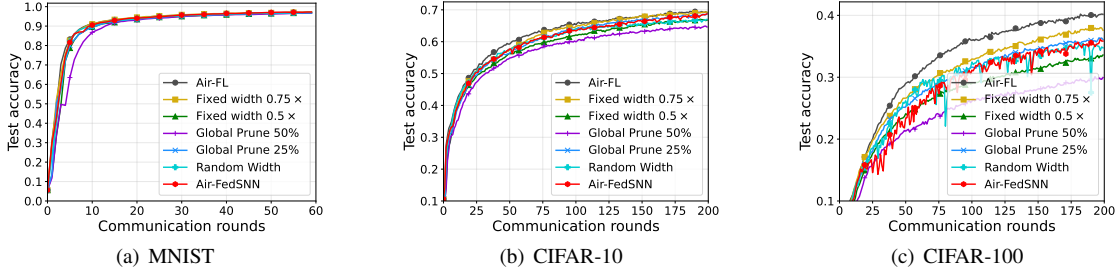


Fig. 3. Test accuracy versus different FL methods.

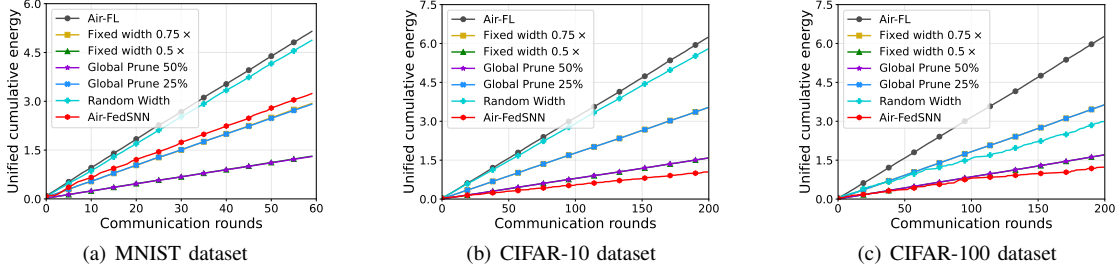


Fig. 4. Unified cumulative energy consumption versus different FL methods.

- **Fixed Width:** All devices use a fixed network width throughout model training. We evaluate the fixed-width settings of 0.5 $\times$  and 0.75 $\times$ .
- **Random Width:** Each device randomly selects a network width at every training round, providing a stochastic width-variation baseline.
- **Global Pruning:** All devices adopt an identical global pruning ratio, defined as the fraction of parameters removed, to ensure signal alignment for AirComp aggregation [14]. We examine pruning ratios of 25% and 50%, matching the retained parameter counts of the Fixed Width method.

#### A. Overall Performance Benchmarking

This section evaluates the learning accuracy and training efficiency of Air-FedSNN against baselines on MNIST, CIFAR-10, and CIFAR-100, demonstrating its superior trade-off between model performance and energy consumption.

Table II summarizes the test accuracy upon convergence and the corresponding average energy consumption per device per round, where the full-width Air-FL is used as the reference (1 $\times$ ). The proposed Air-FedSNN consistently achieves a better balance between accuracy and energy consumption. In the relatively simple MNIST task, our method achieves the highest accuracy of 97.22%, slightly outperforming the full-width Air-FL. On CIFAR-10, it attains 68.72%, comparable

to 69.52% of Air-FL and higher than 67.04% of the Fixed Width 0.5 $\times$  baseline. For the more complex CIFAR-100 dataset, our method outperforms the 0.5 $\times$  baseline (36.17% versus 33.59%), demonstrating robustness on more complex tasks. In terms of energy efficiency, the proposed method reduces energy consumption to 0.45 $\times$  on MNIST. The gain is more pronounced on CIFAR-10, where our method consumes only 1.02 per round compared with 3.82 for Air-FL, and achieves a reduction to approximately 26% of the baseline energy, indicating that dynamic width allocation effectively reduces redundant computation.

The comparative analysis highlights the limitations of existing baselines. Fixed Width methods face a rigid trade-off between accuracy and energy consumption, while Random Width suffers from unstable convergence due to non-adaptive width allocation. Although Global Pruning maintains uniform model structures for AirComp alignment, it lacks flexibility to adapt to heterogeneous device resources. In contrast, Air-FedSNN benefits from dynamic resource allocation. Lyapunov optimization transforms the long-term energy constraint into per-round decisions, while MAB learns device capabilities online, enabling energy-aware and resource-adaptive width allocation without prior knowledge.

Fig. 3 illustrates the convergence of test accuracy across all datasets. It can be observed that Air-FedSNN exhibits a fast convergence rate comparable to Air-FL. Notably, in

TABLE III  
PERFORMANCE COMPARISON OF DIFFERENT FL METHODS WITH DIFFERENT DISTRIBUTIONS.

| FL Methods                | CIFAR-10       |                       |                |                       |             |                       |             |                       | CIFAR-100      |                       |                |                       |             |                       |             |                       |
|---------------------------|----------------|-----------------------|----------------|-----------------------|-------------|-----------------------|-------------|-----------------------|----------------|-----------------------|----------------|-----------------------|-------------|-----------------------|-------------|-----------------------|
|                           | $\alpha = 0.1$ |                       | $\alpha = 0.5$ |                       | $\mu = 0.2$ |                       | $\mu = 0.5$ |                       | $\alpha = 0.1$ |                       | $\alpha = 0.5$ |                       | $\mu = 0.2$ |                       | $\mu = 0.5$ |                       |
|                           | Acc            | Eng                   | Acc            | Eng                   | Acc         | Eng                   | Acc         | Eng                   | Acc            | Eng                   | Acc            | Eng                   | Acc         | Eng                   | Acc         | Eng                   |
| Air-FL                    | 54.86          | 3.63 (1 $\times$ )    | 66.2           | 3.77 (1 $\times$ )    | 69.80       | 3.55 (1 $\times$ )    | 68.22       | 3.83 (1 $\times$ )    | 35.79          | 3.82 (1 $\times$ )    | 39.05          | 3.85 (1 $\times$ )    | 37.59       | 3.63 (1 $\times$ )    | 40.17       | 3.85 (1 $\times$ )    |
| Fixed Width 0.75 $\times$ | 52.95          | 2.05 (0.57 $\times$ ) | 64.51          | 2.14 (0.57 $\times$ ) | 69.38       | 2.01 (0.57 $\times$ ) | 67.04       | 2.17 (0.57 $\times$ ) | 34.22          | 2.21 (0.58 $\times$ ) | 37.59          | 2.23 (0.58 $\times$ ) | 36.49       | 2.10 (0.58 $\times$ ) | 38.05       | 2.23 (0.58 $\times$ ) |
| Fixed Width 0.5 $\times$  | 48.09          | 0.92 (0.25 $\times$ ) | 62.66          | 0.96 (0.25 $\times$ ) | 66.56       | 0.90 (0.25 $\times$ ) | 65.30       | 0.97 (0.25 $\times$ ) | 30.73          | 1.04 (0.27 $\times$ ) | 34.23          | 1.05 (0.27 $\times$ ) | 31.88       | 0.99 (0.27 $\times$ ) | 33.70       | 1.05 (0.27 $\times$ ) |
| Random Width              | 43.87          | 1.68 (0.46 $\times$ ) | 62.93          | 1.76 (0.47 $\times$ ) | 66.76       | 1.46 (0.41 $\times$ ) | 65.23       | 1.53 (0.40 $\times$ ) | 26.64          | 1.82 (0.48 $\times$ ) | 35.25          | 1.76 (0.46 $\times$ ) | 32.91       | 1.76 (0.49 $\times$ ) | 34.87       | 1.86 (0.48 $\times$ ) |
| Global Pruning 25%        | 53.13          | 2.06 (0.57 $\times$ ) | 63.23          | 2.13 (0.57 $\times$ ) | 67.98       | 2.01 (0.57 $\times$ ) | 65.92       | 2.17 (0.57 $\times$ ) | 33.09          | 2.21 (0.58 $\times$ ) | 35.69          | 2.23 (0.58 $\times$ ) | 34.57       | 2.11 (0.58 $\times$ ) | 36.65       | 2.23 (0.58 $\times$ ) |
| Global Pruning 50%        | 49.95          | 0.92 (0.25 $\times$ ) | 59.54          | 0.96 (0.25 $\times$ ) | 63.96       | 0.90 (0.25 $\times$ ) | 61.51       | 0.97 (0.25 $\times$ ) | 27.39          | 1.04 (0.27 $\times$ ) | 30.10          | 1.05 (0.27 $\times$ ) | 28.65       | 0.99 (0.27 $\times$ ) | 30.51       | 1.05 (0.27 $\times$ ) |
| Air-FedSNN (Ours)         | 51.67          | 0.94 (0.26 $\times$ ) | 64.11          | 1.01 (0.27 $\times$ ) | 67.49       | 1.01 (0.28 $\times$ ) | 65.37       | 1.02 (0.26 $\times$ ) | 30.54          | 1.04 (0.27 $\times$ ) | 35.01          | 1.03 (0.27 $\times$ ) | 32.77       | 1.02 (0.28 $\times$ ) | 35.96       | 1.02 (0.26 $\times$ ) |

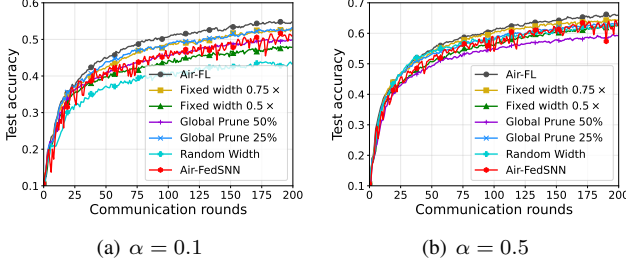


Fig. 5. Test accuracy comparison on CIFAR-10 (Dirichlet distribution).

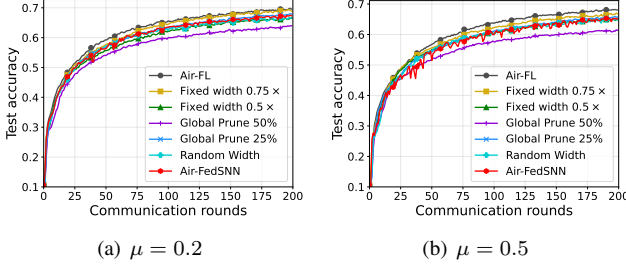


Fig. 7. Test accuracy comparison on CIFAR-10 (Synthetic label skew).

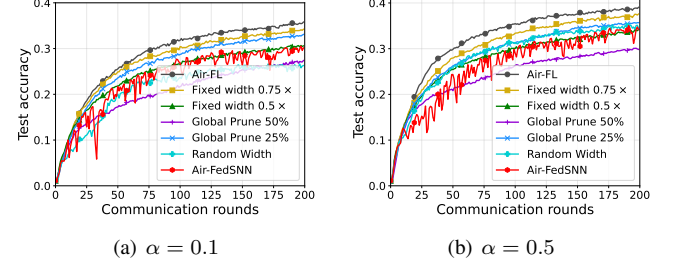


Fig. 6. Test accuracy comparison on CIFAR-100 (Dirichlet distribution).

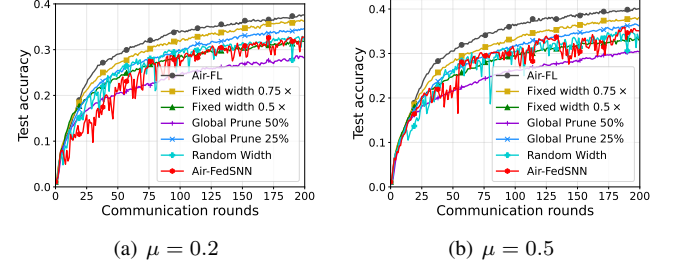


Fig. 8. Test accuracy comparison on CIFAR-100 (Synthetic label skew).

the CIFAR-100 task (Fig. 3(c)), while static methods such as Fixed Width 0.5 $\times$  saturate at a lower accuracy level, our proposed method continues to improve, effectively closing the gap with the full-width model. This confirms that the proposed width-aware aggregation and dynamic scheduling can successfully maintain model capacity even when training involves reduced model widths.

From the cumulative energy perspective shown in Fig. 4, Air-FedSNN achieves the lowest energy consumption among all high-accuracy baselines. Although minor fluctuations appear on MNIST (Fig. 4(a)), this is attributed to the normalized cumulative energy, which is determined by the maximum device energy consumption in each round. Consequently, a device with particularly high instantaneous energy consumption can cause observable fluctuations. Despite these variations, the overall trend confirms the energy efficiency of our method. Furthermore, by reducing the energy budget  $\bar{E}$ , the energy consumption can be further decreased, demonstrating the flexibility of our framework under different resource constraints.

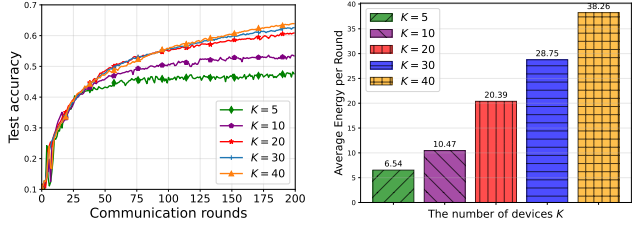
### B. Robustness under Non-IID Data Heterogeneity

This section investigates the performance of Air-FedSNN under non-IID data distributions. While maintaining a balance between energy consumption and accuracy is our primary objective, verifying the method's robustness with heterogeneous local data is crucial. To this end, we simulate two distinct non-IID settings: a Dirichlet distribution with concentration parameters  $\alpha \in \{0.1, 0.5\}$ , and a Pathological non-IID setting

parameterized by  $\mu \in \{0.2, 0.5\}$ . Table III summarizes the accuracy and energy consumption for all FL methods under different non-IID settings, while Figs. 5–8 illustrate the corresponding convergence behavior on CIFAR-10 and CIFAR-100.

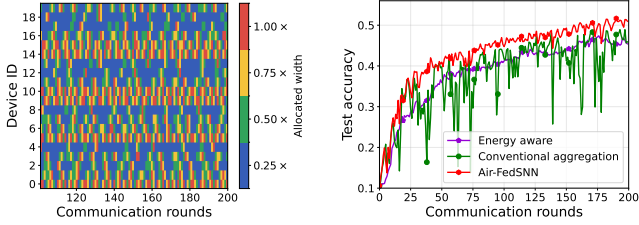
The convergence speed and accuracy of all methods, including the full-width Air-FL, are lower compared to IID scenarios due to statistical heterogeneity of local data. Observing the baselines, static strategies (i.e., Fixed Width and Global Pruning) exhibit noticeable performance degradation, particularly under the Dirichlet setting with  $\alpha = 0.1$  and the pathological case with  $\mu = 0.2$ . This degradation occurs because fixed model capacity or a uniform pruning mask increases the difficulty of learning from skewed data, as these methods lack the flexibility to adapt to diverse local datasets. Meanwhile, Random Width continues to show unstable convergence due to its stochastic width strategy.

In contrast, Air-FedSNN consistently achieves a favorable tradeoff between test accuracy and energy consumption under all heterogeneous settings. As shown in Table III, our approach reaches accuracy comparable to or higher than other compression-based baselines at similar or even lower energy budgets. Notably, when compared with low-energy methods such as Fixed Width 0.5 $\times$  and Global Pruning 50%, Air-FedSNN achieves higher test accuracy across different data distribution cases on both datasets. This gain is also reflected in the convergence curves (Figs. 5–8), where Air-FedSNN shows higher accuracy and faster convergence. This indicates that the proposed dynamic width allocation and



(a) Test accuracy (b) Average energy cost per Round  $t$

Fig. 9. Performance evaluation with varying numbers of devices.



(a) Width allocation distribution (b) Model performance comparison

Fig. 11. Width allocation distribution and model performance analysis.

aggregation mechanism is effective not only in standard settings but also remains robust under severe data heterogeneity.

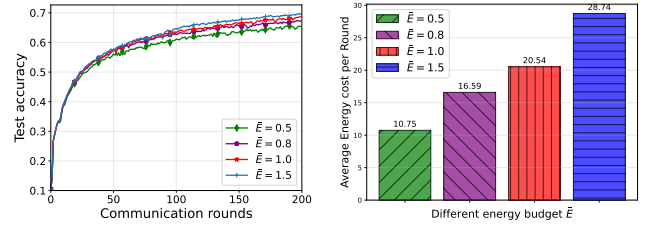
### C. Parameter Sensitivity and Configuration Tradeoffs

This section verifies the adaptability of Air-FedSNN by evaluating its performance under varying configurations of two critical system parameters: the number of participating devices ( $K$ ) and the average per-device energy budget ( $\bar{E}$ ), using the CIFAR-10 dataset for all experiments.

We first investigate the influence of the number of devices on the model performance. To ensure a fair comparison, we allocate a fixed dataset size of 1,000 samples to each device. As illustrated in Fig. 9(a), increasing  $K$  from 5 to 40 results in a consistent improvement in test accuracy. This performance enhancement is attributed to the increase in the total volume of training data available globally, which improves the generalization capability of the aggregated model. However, this gain in accuracy comes at the cost of increased system energy consumption. As shown in Fig. 9(b), the average energy cost per round increases as more devices are scheduled.

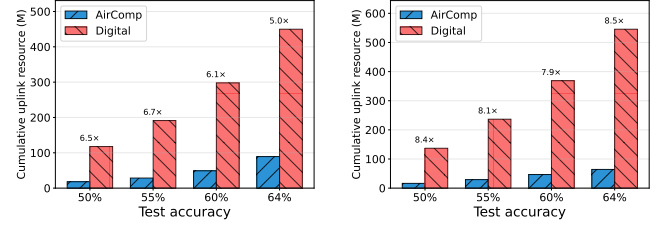
Subsequently, we examine system performance under varying energy constraints by adjusting the average energy budget  $\bar{E}$ . Fig. 10 presents the test accuracy and corresponding average energy cost under different  $\bar{E}$  ranging from 0.5 to 1.5. It is observed that relaxing the energy budget leads to faster convergence and higher final test accuracy. This acceleration occurs because a higher energy allowance enables the dynamic scheduler to select wider sub-networks more frequently, thereby increasing effective model capacity. Consequently, as depicted in Fig. 10(b), the system energy consumption per round increases proportionally with the preset budget  $\bar{E}$ . This confirms that our framework effectively leverages available energy resources to maximize learning performance.

To further analyze device heterogeneity, we conduct an additional sensitivity evaluation by varying the computation heterogeneity level. The baseline CPU frequency is sampled



(a) Test accuracy (b) Average energy cost per Round  $t$

Fig. 10. Performance evaluation with different energy budgets.



(a)  $\bar{E} = 1.0$  J

(b)  $\bar{E} = 1.5$  J

Fig. 12. Uplink channel uses required by AirComp and digital aggregation.

TABLE IV  
IMPACT OF COMPUTATION HETEROGENEITY ON AIR-FEDSNN.

| Heterogeneity level | Accuracy | Energy | Average width |
|---------------------|----------|--------|---------------|
| Low                 | 68.25    | 1.02   | 0.471         |
| Medium              | 68.72    | 1.02   | 0.488         |
| High                | 68.98    | 1.00   | 0.500         |

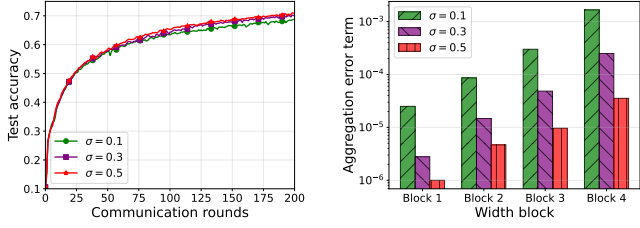
from  $\bar{f}_k \sim \mathcal{U}(3, 4)$  GHz,  $\mathcal{U}(2, 5)$  GHz, and  $\mathcal{U}(1, 6)$  GHz, corresponding to low, medium, and high heterogeneity, respectively. These settings share the same mean CPU frequency of 3.5 GHz but have gradually increasing variances.

As shown in Table IV, Air-FedSNN maintains stable accuracy and energy consumption across different computation heterogeneity levels. The average selected width slightly increases with stronger heterogeneity, indicating that the proposed strategy can exploit resource diversity by assigning larger widths to more capable devices.

### D. Width Distribution Analysis

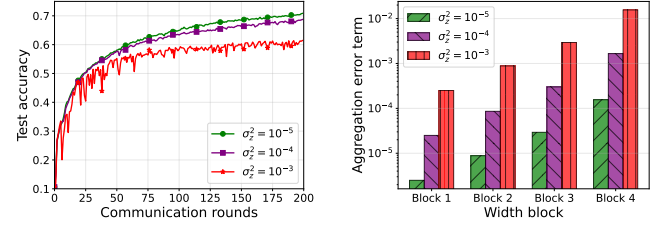
A potential concern is that some devices may repeatedly select large widths during training, which could make the global model overly influenced by a subset of devices and thus lead to biased learning under non-IID data. To examine this issue, we analyze the width allocation distribution and model performance under the non-IID Dirichlet setting with  $\alpha = 0.1$ , as shown in Fig. 11.

From Fig. 11(a), the allocated model widths of devices continue to change over the last 100 communication rounds rather than converging to a fixed width, indicating that the proposed mechanism maintains dynamic width adaptation. This is mainly attributed to the Lyapunov energy queue, where frequent large-width selection increases energy consumption and enlarges the virtual queue, thereby discouraging persistent large-width allocation. Fig. 11(b) compares model performance under two baselines: Energy-aware aggregation, where devices greedily select the maximum feasible width based on instantaneous energy without



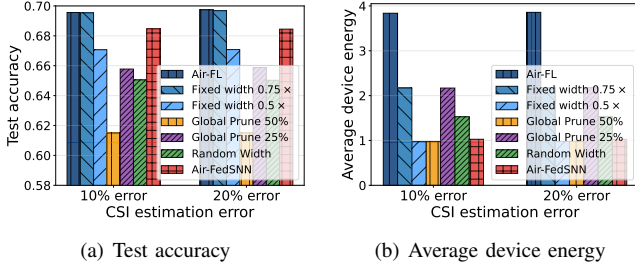
(a) Test accuracy (b) Aggregation error performance

Fig. 13. Performance analysis under different power control factors.



(a) Test accuracy (b) Aggregation error performance

Fig. 14. Performance analysis under different noise variances.



(a) Test accuracy (b) Average device energy

Fig. 15. Training performance comparison under CSI estimation error.

long-term constraint, and Conventional aggregation, which ignores width heterogeneity and directly aggregates local updates using standard FL averaging. The results show that Air-FedSNN achieves higher and more stable accuracy, demonstrating its effectiveness in mitigating biased learning under heterogeneous settings.

### E. Communication Efficiency Evaluation of AirComp

To further demonstrate the efficiency of AirComp, we compare it with a digital aggregation baseline under different energy budgets, as shown in Fig. 12. The digital baseline requires each device to separately upload its local updates, resulting in a communication cost that scales linearly with the number of devices. In contrast, AirComp enables simultaneous transmission within each active width block, significantly reducing uplink communication overhead by exploiting the signal superposition property of wireless channels.

As shown in Fig. 12, AirComp consistently requires substantially fewer uplink resources than digital aggregation across all target accuracy levels. Moreover, the performance gain becomes more pronounced as the energy budget increases. This is because higher energy budgets activate more devices and larger-width blocks, where AirComp can better exploit concurrent transmissions, while the digital scheme suffers from proportional growth in communication cost. Overall, the results verify that AirComp provides a scalable communication-efficient aggregation mechanism, especially in large-scale and energy-abundant settings.

### F. Impact of Communication Conditions

This subsection investigates the impact of communication conditions on the proposed Air-FedSNN framework, including power control factors, channel noise, and imperfect CSI.

As shown in Fig. 13 and Fig. 14, a larger  $\sigma$  or a lower noise variance  $\sigma_z^2$  effectively reduces aggregation error by

improving channel inversion accuracy and mitigating channel noise, thereby yielding more reliable global model updates. Moreover, the impact of these communication parameters is not uniform across width blocks. Higher-width blocks are more sensitive to channel variations due to fewer participating devices, which results in more pronounced aggregation errors under identical wireless conditions.

To further evaluate robustness under imperfect CSI, we consider channel estimation errors of 10% and 20%. As shown in Fig. 15, increasing CSI estimation error leads to a degradation in test accuracy, while the average device energy remains relatively stable. This is because channel mismatch introduces additional aggregation noise that affects model convergence. Nevertheless, the proposed Air-FedSNN consistently achieves the best tradeoff between accuracy and energy under different CSI error levels. In particular, compared with baseline methods, Air-FedSNN maintains a better accuracy-energy tradeoff, demonstrating that the joint design of width adaptation and Lyapunov-based energy control effectively mitigates the impact of communication uncertainty in AirComp aggregation.

### G. Extension to Larger models and Tasks

To evaluate the scalability of the proposed method on larger models and more complex tasks, we conduct experiments on the Tiny-ImageNet dataset. Tiny-ImageNet is a 200-class subset of ImageNet with  $64 \times 64$  images. Compared with CIFAR-10 and CIFAR-100 benchmarks, it contains more similar classes and larger intra-class variations, making it a more challenging classification task. On the Tiny-ImageNet dataset, we further consider two representative models, i.e., CNN and MobileNetV2, which is designed for mobile devices and widely used in edge networks. For MobileNetV2 [36], batch normalization is replaced with group normalization to improve training stability under heterogeneous local models. Compared with CNN, MobileNetV2 has a larger parameter scale and higher model complexity. Therefore, to account for the increased computational cost, the per-sample CPU cycles and average energy budget are set to  $6 \times 10^6$  and  $\bar{E} = 12$ . For MobileNetV2 on Tiny-ImageNet, all methods are trained for 200 communication rounds with an initial learning rate of 0.1 and a decay factor of 0.993 to accelerate convergence.

Table V shows the performance comparison between different baselines under CNN and MobileNetV2 models. MobileNetV2 achieves better performance at the cost of higher energy consumption. With CNN model, Air-FedSNN re-

TABLE V  
PERFORMANCE COMPARISON ON TINY-IMAGENET DATASET.

| FL methods         | CNN      |        | MobileNetV2 |        |
|--------------------|----------|--------|-------------|--------|
|                    | Accuracy | Energy | Accuracy    | Energy |
| Air-FL             | 25.03    | 1×     | 38.98       | 1×     |
| Fixed Width 0.75×  | 22.06    | 0.59×  | 38.17       | 0.59×  |
| Fixed Width 0.5×   | 21.13    | 0.29×  | 34.82       | 0.29×  |
| Random Width       | 22.44    | 0.49×  | 38.09       | 0.49×  |
| Global Pruning 25% | 20.87    | 0.59×  | 29.48       | 0.59×  |
| Global Pruning 50% | 16.21    | 0.29×  | 28.33       | 0.29×  |
| Air-FedSNN (Ours)  | 22.90    | 0.30×  | 37.67       | 0.32×  |

duces energy consumption to 0.30× while maintaining comparable accuracy (22.90%) compared with Air-FL (25.03%), corresponding to a 2.13% accuracy drop with a 70% energy reduction. Similarly, with MobileNetV2, it achieves 37.67% accuracy, close to Air-FL (38.98%), while reducing energy consumption from 1× to 0.32×, corresponding to a 68% energy saving with a 1.31% accuracy degradation. Across different settings, Air-FedSNN consistently achieves better accuracy under comparable energy levels. Compared with Fixed Width 0.5× and Global Pruning 50% (0.29× energy) in the MobileNetV2 setting, it achieves higher accuracy of 37.67%, improving them by 2.85 and 9.34 percentage points, respectively, demonstrating its training efficiency.

These results indicate that Air-FedSNN achieves a better balance between training efficiency and performance compared with baselines. In particular, it maintains better accuracy under substantial energy reduction, highlighting its effectiveness and scalability for more complex tasks and models.

## VII. CONCLUSION

This paper introduces the Air-FedSNN framework, combining AirComp and SNNs to enhance communication, computation, and energy efficiency in heterogeneous IoT networks. A width-aware aggregation mechanism is developed to improve aggregation performance for width-varying local models, and the convergence analysis is provided to illustrate the impact of network widths variability and aggregation error on learning performance. Additionally, a dynamic width allocation strategy based on Lyapunov and MAB theory is proposed to optimize energy usage and model performance without prior knowledge of device capabilities. Extensive simulations on MNIST, CIFAR-10 and CIFAR-100 datasets confirm that Air-FedSNN achieves superior accuracy-energy tradeoffs compared to conventional FL approaches, offering an effective solution for resource-constrained IoT environments.

Several future directions remain, including: 1) introducing knowledge distillation from the full-width global model to local sub-models to enhance training performance; 2) jointly optimizing model width and wireless resource allocation to mitigate AirComp aggregation errors under time-varying channel conditions and heterogeneous device capabilities.

## APPENDIX A PROOF OF THEOREM 1

Since  $\beta_k^t \in (0, 1]$ , we have  $1 < e^{\beta_k^t} \leq e$ , and thus the upper bound of  $\rho_{k,j}^t$  satisfies  $\rho_{k,j}^t \leq \frac{e}{S_j^t} \leq e$ , where  $S_j^t$  denotes the number of devices participating in width block  $j$  at round  $t$ .

From eq. (13), the full-width global model update can be written as

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \eta \sum_{k=1}^K \sum_{l=0}^{\Gamma-1} \nabla \tilde{F}_k(\mathbf{w}_k^{t,l}) + \tilde{\mathbf{z}}^t, \quad (45)$$

where the concatenated gradient contributed by device  $k$  is

$$\nabla \tilde{F}_k(\mathbf{w}_k^{t,l}) = \text{concat}_{j=1}^J \left( \rho_{k,j}^t \nabla F_{k,j}(\mathbf{w}_k^{t,l}) \right), \quad (46)$$

and the aggregated perturbation is  $\tilde{\mathbf{z}}^t = \text{concat}_{j=1}^J \tilde{\mathbf{z}}_j^t$ .

Next, we will present the proof of **Theorem 1** which illustrates the convergence performance of Air-FedSNN. According to **Assumption 1** and eq. (45), we have

$$\begin{aligned} & \mathbb{E} [F(\mathbf{w}^{t+1}) - F(\mathbf{w}^t)] \\ & \leq \underbrace{\eta \sum_{l=0}^{\Gamma-1} \mathbb{E} \left\langle \nabla F(\mathbf{w}^t), -\sum_{k=1}^K \nabla \tilde{F}_k(\mathbf{w}_k^{t,l}) \right\rangle}_{A_1} \\ & \quad + \underbrace{\frac{L\eta^2}{2} \mathbb{E} \left\| \sum_{k=1}^K \sum_{l=0}^{\Gamma-1} \nabla \tilde{F}_k(\mathbf{w}_k^{t,l}) \right\|^2}_{A_2} + \underbrace{\frac{L}{2} \mathbb{E} \|\tilde{\mathbf{z}}^t\|^2}_{A_3}. \end{aligned} \quad (47)$$

Then, we derive the upper bound of the term  $A_1$  as follows

$$\begin{aligned} A_1 & = -\mathbb{E} \left\langle \nabla F(\mathbf{w}^t), \sum_{k=1}^K \nabla \tilde{F}_k(\mathbf{w}_k^{t,l}) \right\rangle \\ & \stackrel{(a)}{\leq} \frac{1}{2} \sum_{j=1}^J \|\nabla F_j(\mathbf{w}^t)\|^2 - \sum_{j=1}^J \mathbb{E} \left\langle \nabla F_j(\mathbf{w}^t), \sum_{k \in \mathcal{K}} \rho_{k,j}^t \nabla F_{k,j}(\mathbf{w}^t) \right\rangle \\ & \quad + \frac{1}{2} \sum_{j=1}^J \sum_{k \in \mathcal{K}} \rho_{k,j}^t \mathbb{E} \left\| \nabla F_{k,j}(\mathbf{w}_k^{t,l}) - \nabla F_{k,j}(\mathbf{w}^t) \right\|^2 \\ & \stackrel{(b)}{\leq} -\frac{1}{2} \sum_{j=1}^J \|\nabla F_j(\mathbf{w}^t)\|^2 \\ & \quad + \sum_{j=1}^J \frac{e}{2S_j^t} \sum_{k \in \mathcal{K}} \mathbb{E} \left\| \nabla F_{k,j}(\mathbf{w}_k^{t,l}) - \nabla F_{k,j}(\mathbf{w}^t) \right\|^2 \end{aligned} \quad (48)$$

where (a) follows the inequality  $-a^T b \leq \frac{\|a\|^2}{2x} + \frac{x\|b\|^2}{2}$ , and (b) follows eq. (18) in **Assumption 2**.

Recalling that  $\hat{S} \leq S_j^t$  denotes the minimum number of devices participating in width block  $j$ . Then, by applying **Assumption 1** and **Lemma 1**, we can obtain

$$\begin{aligned} A_1 & \leq -\frac{1}{2} \sum_{j=1}^J \|\nabla F_j(\mathbf{w}^t)\|^2 + \frac{eL^2}{2\hat{S}} \sum_{k \in \mathcal{K}} \mathbb{E} \left\| \mathbf{w}_k^{t,l} - \mathbf{w}^t \right\|^2 \\ & \leq -\frac{1}{2} \|\nabla F(\mathbf{w}^t)\|^2 + \frac{eL^2\eta^2 K \Gamma^2 G^2}{\hat{S}} + \frac{eL^2 K \varphi^2}{\hat{S}} \|\mathbf{W}\|^2. \end{aligned} \quad (49)$$

Next, we derive the upper bound of the term  $A_2$  as follows

$$\begin{aligned}
A_2 &= \mathbb{E} \left\| \sum_{k=1}^K \sum_{l=0}^{\Gamma-1} \nabla \tilde{F}_k \left( \mathbf{w}_k^{t,l} \right) \right\|^2 \\
&= \sum_{j=1}^J \mathbb{E} \left\| \sum_{k=1}^K \sum_{l=0}^{\Gamma-1} \rho_{k,j}^t \nabla F_{k,j} \left( \mathbf{w}_k^{t,l} \right) \right\|^2 \\
&\stackrel{(a)}{\leq} \frac{2e\Gamma}{\hat{S}_j^t} \sum_{l=0}^{\Gamma-1} \sum_{j=1}^J \sum_{k=1}^K \mathbb{E} \left\| \nabla F_{k,j} \left( \mathbf{w}_k^{t,l} \right) - \nabla F_{k,j} \left( \mathbf{w}^t \right) \right\|^2 \\
&\quad + 2\Gamma^2 \sum_{j=1}^J \mathbb{E} \left\| \sum_{k=1}^K \rho_{k,j}^t \nabla F_{k,j} \left( \mathbf{w}^t \right) \right\|^2
\end{aligned} \tag{50}$$

where (a) follows the inequality  $\|a + b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$ . By applying **Assumption 1**, **2** and **Lemma 1**, we can get

$$\begin{aligned}
A_2 &\leq \frac{2e\Gamma^2 L^2}{\hat{S}} \sum_{k=1}^K \mathbb{E} \left\| \mathbf{w}_k^{t,l} - \mathbf{w}^t \right\|^2 + 2\Gamma^2 \sum_{j=1}^J \mathbb{E} \left\| \nabla F_j \left( \mathbf{w}^t \right) \right\|^2 \\
&\leq \frac{4eK\eta^2 \Gamma^4 L^2 G^2}{\hat{S}} + \frac{4eK\Gamma^2 L^2 \varphi^2}{\hat{S}} \|\mathbf{W}\|^2 \\
&\quad + 2\Gamma^2 \mathbb{E} \left\| \nabla F \left( \mathbf{w}^t \right) \right\|^2,
\end{aligned} \tag{51}$$

Then, by substituting the results of eq. (49) and eq. (51) into eq. (47), and applying  $A_3 = \mathbb{E} \|\tilde{\mathbf{z}}^t\|^2 \leq \sum_{j=1}^J \frac{\sigma_z^2}{(S_j^t \sigma_j^t)^2}$ , we can obtain

$$\begin{aligned}
\mathbb{E} [F(\mathbf{w}^{t+1}) - F(\mathbf{w}^t)] &\leq \left( L\eta^2 \Gamma^2 - \frac{\eta\Gamma}{2} \right) \|\nabla F(\mathbf{w}^t)\|^2 \\
&\quad + \frac{e\eta^3 L^2 K \Gamma^3 + 2e\eta^4 K \Gamma^4 L^3}{\hat{S}} G^2 + \frac{L}{2} \sum_{j=1}^J \frac{\sigma_z^2}{(S_j^t \sigma_j^t)^2} \\
&\quad + \frac{e\eta\Gamma L^2 K + 2e\eta^2 K \Gamma^2 L^3}{\hat{S}} \varphi^2 \|\mathbf{W}\|^2.
\end{aligned} \tag{52}$$

For any  $\mathbf{w}^t \in \mathbb{R}^q$  at round  $t \in \{1, \dots, T\}$ , there existing  $F(\mathbf{w}^t) \geq F^*$ . Therefore, by summing up eq. (52) for all  $T$  rounds and performing simple algebraic operations, the convergence upper bound after  $T$  rounds is

$$\begin{aligned}
\frac{1}{T} \sum_{t=1}^T \mathbb{E} [\|\nabla F(\mathbf{w}^t)\|^2] &\leq \frac{2}{T\eta\Gamma(1-2L\eta\Gamma)} \mathbb{E} [F(\mathbf{w}^1) - F^*] \\
&\quad + \frac{2eL^2 K \eta^2 \Gamma^2 + 4eL^3 K \eta^3 \Gamma^3}{\hat{S}(1-2L\eta\Gamma)} G^2 \\
&\quad + \frac{2eL^2 K + 4eL^3 K \eta\Gamma}{\hat{S}(1-2L\eta\Gamma)} \varphi^2 \|\mathbf{W}\|^2 \\
&\quad + \frac{L}{T(\eta\Gamma - 2L\eta^2 \Gamma^2)} \sum_{t=1}^T \sum_{j=1}^J \frac{\sigma_z^2}{(S_j^t \sigma_j^t)^2}
\end{aligned} \tag{53}$$

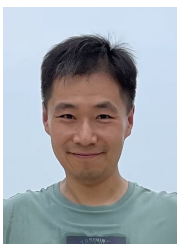
## REFERENCES

- [1] Y. Li, F. Li, S. Yang, C. Zhang, L. Zhu, and Y. Wang, "A cooperative analysis to incentivize communication-efficient federated learning," *IEEE Trans. Mob. Comput.*, vol. 23, no. 10, pp. 10175–10190, Oct. 2024.
- [2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proc. 20th Int. Conf. Artif. Intell. Stat. (AISTATS)*, vol. 54, Apr. 2017, pp. 1273–1282.
- [3] J. Du, B. Jiang, C. Jiang, Y. Shi, and Z. Han, "Gradient and channel aware dynamic scheduling for over-the-air computation in federated edge learning systems," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 4, pp. 1035–1050, Apr. 2023.
- [4] M. Ye, X. Fang, B. Du, P. C. Yuen, and D. Tao, "Heterogeneous federated learning: State-of-the-art and research challenges," *ACM Comput. Surv.*, vol. 56, no. 3, pp. 79:1–79:44, Oct. 2023.
- [5] A. Sahin and R. Yang, "A survey on over-the-air computation," *IEEE Commun. Surv. Tutorials*, vol. 25, no. 3, pp. 1877–1908, 3rd Quart. 2023.
- [6] X. Cao, Z. Lyu, G. Zhu, J. Xu, L. Xu, and S. Cui, "An overview on over-the-air federated edge learning," *IEEE Wirel. Commun.*, vol. 31, no. 3, pp. 202–210, Jun. 2024.
- [7] G. Zhu, Y. Du, D. Gündüz, and K. Huang, "One-bit over-the-air aggregation for communication-efficient federated edge learning: Design and convergence analysis," *IEEE Trans. Wirel. Commun.*, vol. 20, no. 3, pp. 2120–2135, Mar. 2021.
- [8] X. Fan, Y. Wang, Y. Huo, and Z. Tian, "1-bit compressive sensing for efficient federated learning over the air," *IEEE Trans. Wirel. Commun.*, vol. 22, no. 3, pp. 2139–2155, Mar. 2023.
- [9] M. M. Amiri and D. Gündüz, "Federated learning over wireless fading channels," *IEEE Trans. Wirel. Commun.*, vol. 19, no. 5, pp. 3546–3557, May 2020.
- [10] J. Ahn, M. Bennis, and J. Kang, "Model compression via pattern shared sparsification in analog federated learning under communication constraints," *IEEE Trans. Green Commun. Netw.*, vol. 7, no. 1, pp. 298–312, Mar. 2023.
- [11] K. Xu, H. H. Yang, Z. Zhao, W. Hong, T. Q. S. Quek, and M. Peng, "Pruning analog over-the-air distributed learning models with accuracy loss guarantee," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Seoul, Korea, May 2022, pp. 5202–5207.
- [12] F. M. A. Khan, H. Abou-Zeid, and S. A. Hassan, "Deep compression for efficient and accelerated over-the-air federated learning," *IEEE Internet Things J.*, vol. 11, no. 15, pp. 25802–25817, Aug. 2024.
- [13] H. Xing, O. Simeone, and S. Bi, "Federated learning over wireless device-to-device networks: Algorithms and convergence analysis," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3723–3741, Dec. 2021.
- [14] Z. Zhao, K. Xu, W. Hong, M. Peng, Z. Ding, T. Q. S. Quek, and H. H. Yang, "Model pruning for distributed learning over the air," *IEEE Trans. Signal Process.*, vol. 72, pp. 5533–5549, 2024.
- [15] Z. Wang, K. Huang, and Y. C. Eldar, "Spectrum breathing: A spectrum-efficient method for protecting over-the-air federated learning against interference," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Kuala Lumpur, Malaysia, Dec. 2023, pp. 4952–4957.
- [16] J. Yu, L. Yang, N. Xu, J. Yang, and T. S. Huang, "Slimmable neural networks," in *Proc. 7th Int. Conf. Learn. Represent. (ICLR)*, New Orleans, LA, USA, May 2019.
- [17] G. Zhu, J. Xu, K. Huang, and S. Cui, "Over-the-air computing for wireless data aggregation in massive iot," *IEEE Wirel. Commun.*, vol. 28, no. 4, pp. 57–65, Aug. 2021.
- [18] Y. Liang, Q. Chen, G. Zhu, H. Jiang, Y. C. Eldar, and S. Cui, "Communication-and-energy efficient over-the-air federated learning," *IEEE Trans. Wirel. Commun.*, vol. 24, no. 1, pp. 767–782, Jan. 2025.
- [19] Y. Sun, S. Zhou, Z. Niu, and D. Gündüz, "Dynamic scheduling for over-the-air federated edge learning with energy constraints," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 227–242, Jan. 2022.
- [20] Q. An, Y. Zhou, Z. Wang, H. Shan, Y. Shi, and M. Bennis, "Online optimization for over-the-air federated learning with energy harvesting," *IEEE Trans. Wirel. Commun.*, vol. 23, no. 7, pp. 7291–7306, Jul. 2024.
- [21] W. J. Yun, Y. Kwak, H. Baek, S. Jung, M. Ji, M. Bennis, J. Park, and J. Kim, "Slimfl: Federated learning with superposition coding over slimmable neural networks," *IEEE/ACM Trans. Netw.*, vol. 31, no. 6, pp. 2499–2514, Dec. 2023.
- [22] H. Baek, G. S. Kim, S. Park, A. F. Molisch, and J. Kim, "Slimmable federated reinforcement learning for energy-efficient proactive caching," *IEEE Trans. Netw.*, vol. 33, no. 4, pp. 2079–2094, Aug. 2025.
- [23] Y. Xu, Y. Liao, H. Xu, Z. Wang, L. Wang, J. Liu, and C. Qian, "Fednsn: Training slimmable neural network with federated learning in edge computing," *IEEE Trans. Netw.*, vol. 33, no. 1, pp. 414–429, Feb. 2025.
- [24] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," in *Proc. 8th Int. Conf. Learn. Represent. (ICLR)*, Addis Ababa, Ethiopia, Apr. 2020.

- [25] E. Diao, J. Ding, and V. Tarokh, "Heterofl: Computation and communication efficient federated learning for heterogeneous clients," in *Proc. 9th Int. Conf. Learn. Represent. (ICLR), Virtual Event, Austria*, May 2021.
- [26] H. Zhou, T. Lan, G. Venkataramani, and W. Ding, "Every parameter matters: Ensuring the convergence of federated learning with dynamic heterogeneous models reduction," in *Proc. 37th Conf. Neural Inf. Process. Syst. (NeurIPS), New Orleans, LA, USA*, Dec. 2023.
- [27] X. Cao, G. Zhu, J. Xu, and S. Cui, "Transmission power control for over-the-air federated averaging at network edge," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 5, pp. 1571–1586, May 2022.
- [28] Y. Zou, Z. Wang, X. Chen, H. Zhou, and Y. Zhou, "Knowledge-guided learning for transceiver design in over-the-air federated learning," *IEEE Trans. Wirel. Commun.*, vol. 22, no. 1, pp. 270–285, Jan. 2023.
- [29] X. Lian, Y. Huang, Y. Li, and J. Liu, "Asynchronous parallel stochastic gradient for nonconvex optimization," in *Proc. 28th Conf. Neural Inf. Process. Syst. (NeurIPS), Montreal, QC, Canada*, Dec. 2015, pp. 2737–2745.
- [30] J. Wang and G. Joshi, "Cooperative SGD: A unified framework for the design and analysis of communication-efficient SGD algorithms," *arXiv:1808.07576*, 2018.
- [31] C. Hu, Z. Chen, and E. G. Larsson, "Energy-efficient federated edge learning with streaming data: A lyapunov optimization approach," *IEEE Trans. Commun.*, vol. 73, no. 2, pp. 1142–1156, Feb. 2025.
- [32] X. Liu, Y. Deng, and T. Mahmoodi, "Wireless distributed learning: A new hybrid split and federated learning approach," *IEEE Trans. Wirel. Commun.*, vol. 22, no. 4, pp. 2650–2665, Apr. 2023.
- [33] J. Zhang, Y. Hua, H. Wang, T. Song, Z. Xue, R. Ma, and H. Guan, "Fedala: Adaptive local aggregation for personalized federated learning," in *Proc. 37th AAAI Conf. Artif. Intell. (AAAI), Washington, DC, USA*, Feb. 2023, pp. 11 237–11 244.
- [34] Q. Li, Y. Diao, Q. Chen, and B. He, "Federated learning on non-iid data silos: An experimental study," in *Proc. 38th IEEE Int. Conf. Data Eng. (ICDE), Kuala Lumpur, Malaysia*, May 2022, pp. 965–978.
- [35] W. Li, Y. Peng, M. Du, F. Sun, X. Wang, and L. Shen, "Hypernetwork aggregation for decentralized personalized federated learning," in *Proc. 34th Int. Joint Conf. Artif. Intell. (IJCAI), Montreal, Canada*, Aug. 2025, pp. 1440–1448.
- [36] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Salt Lake City, UT, USA*, Jun. 2018, pp. 4510–4520.



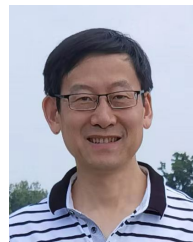
**Yue Zhang** received the M.S. degree from China University of Mining and Technology, Xuzhou, China, in 2021, where he is currently working toward the Ph.D. degree. His research interests include mobile edge computing, machine learning, and wireless communications.



**Guopeng Zhang** received the Ph.D. degree from the School of Communication Engineering, Xidian University, Xi'an, China, in 2009. He is currently a Professor with the School of Computer Science and Technology, China University of Mining and Technology, Xuzhou, China. He has authored or coauthored more than 60 journal and conference papers. His main research interests include distributed machine learning and mobile edge computing.



**Kezhi Wang** received the Ph.D. degree in engineering from the University of Warwick, U.K. He was with the University of Essex and Northumbria University, U.K. Currently, he is a Senior Lecturer with the Department of Computer Science, Brunel University London, U.K. His research interests include wireless communications, mobile edge computing, and machine learning.



**Kun Yang** received his PhD from the Department of Electronic & Electrical Engineering of University College London (UCL), UK. He is currently a Chair Professor of Nanjing University and an affiliated professor at the University of Essex. His main research interests include wireless networks and communications, communication-computing cooperation, and AI (artificial intelligence) for wireless. He has published 500+ papers and filed 50 patents. He serves on the editorial boards of a few IEEE journals (e.g., IEEE WCM, TVT, TNB). He is a Deputy Editor-in-Chief of IET Smart Cities Journal. He has been a Judge of GSMA GLOMO Award at World Mobile Congress – Barcelona since 2019. He was a Distinguished Lecturer of IEEE ComSoc, a Recipient of the 2024 IET Achievement Medals and the Recipient of 2024 IEEE CommSoft TC's Technical Achievement Award. He is a Member of Academia Europaea (MAE), a Fellow of IEEE, a Fellow of IET and a Distinguished Member of ACM.