This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI: 10.1109/TMC.2025.3582683, IEEE Transactions on Mobile Computing

Enhancing Collaborative Machine Learning in Resource-Limited Networks through Knowledge Distillation and Over-the-Air Computation

Yue Zhang, Guopeng Zhang, Kun Yang, Yao Wen, and Kezhi Wang

***Abstract*—Conventional collaborative machine learning (CML) faces significant challenges in resource-constrained environments, such as emergency scenarios with limited power, bandwidth, and computing resources, leading to increased communication delays and energy consumption. To address these issues, this paper introduces `Air-CoKD`, a novel CML framework designed to reduce resource consumption and training latency while preserving model performance. `Air-CoKD` leverages knowledge distillation (KD) to minimize data transmission by avoiding the direct sharing of model parameters. It also integrates over-the-air computation (AirComp) to aggregate local logits, optimizing bandwidth utilization. To address the dimensional differences in local logits caused by the unbalanced device data class, `Air-CoKD` employs orthogonal frequency division multiplexing (OFDM) to transmitting local logits for different target classes. To handle aggregation errors introduced by AirComp, we conduct a detailed analysis of error bounds. Specifically, we convert the Kullback-Leibler (KL) divergence, used in KD loss function, into a quadratic upper bound for precise error quantification and effective optimization. Based on these insights, we propose a strategy to manage bandwidth constraints, transmission power limits, and device energy budgets within `Air-CoKD`. Extensive simulations demonstrate that `Air-CoKD` surpasses state-of-the-art methods, effectively balancing training efficiency and model performance. The framework proves to be a robust solution for CML in resource-constrained networks.**

***Index Terms*—Collaborative machine learning, resource-constrained networks, over-the-air computation, knowledge distillation, convergence analysis.**

I. Introduction

Collaborative machine learning (CML) [1] involves multiple devices working together to train AI models by sharing data, model parameters, or gradients. Federated learning (FL) [2] is a specialized instance of CML that focuses on preserving data privacy by keeping data local to devices and only exchanging model parameters or gradients. For example, `FedAvg` [3] aggregates model parameters, while `FedSGD` [4] relies on aggregating gradients. Despite the advancements in CML, deploying these methods in resource-constrained environments [5], [6], [7], such as during emergency rescue operations [8], presents significant challenges. Natural disasters like earthquakes and floods disrupt power supplies and communication networks [9], complicating the management of frequent model parameter or gradient updates. This repeated transmission strains limited bandwidth and exacerbates energy shortages, leading to increased training delays and hindering timely decision-making. The situation becomes even more challenging in multi-modal learning scenarios, where diverse data types (e.g., images, text, and sensor readings) and larger training model further intensify communication and computation demands [10], [11].

To address these issues, we propose a novel CML method that aims to reduce resource consumption and latency while maintaining acceptable model performance. Our approach includes two key innovations: (1) We replace traditional parameter or gradient aggregation with knowledge aggregation, also called knowledge distillation (KD), through a federated knowledge distillation (`FedKD`) approach [12]. `FedKD` involves transmitting only the probability distributions of target classes, or *local logits*, which significantly lowers data transmission, energy consumption, and training delays. (2) We use Over-the-Air Computation (AirComp) [13] to compute the *global logits* needed for `FedKD`. AirComp enables simultaneous transmission of local logits across all available channel resources, optimizing bandwidth usage and improving computational efficiency by directly aggregating local logits into global logits.

Our proposed CML framework, `Air-CoKD`, leverages these innovations to optimize channel resource utilization, reduce device energy consumption, and minimize training delays while maintaining high model performance. However, balancing these goals presents several challenges that need to be addressed:

- *Dimension inconsistencies in AirComp-based logit aggregation:* In distributed environments, non-independent and identically distributed (non-IID) sampling often results in devices holding data with distinct class distributions, causing significant dimensional differences in locally generated logits. Since AirComp requires aligned dimensions for efficient global logit aggregation, this data imbalance introduces inconsistencies that hinder seamless aggregation [14].
- *Uncertain impact of logits aggregation error:* Unlike `FedAvg` or `FedSGD`, which directly enhance model performance by sharing model parameters or gradients, `Air-CoKD` employs KD regularizers to indirectly influence local training. This indirect influence makes it

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

Kun Yang is with the School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, U.K. (e-mail: kunyang@essex.ac.uk).

Yao Wen is with the School of Intelligent Software and Engineering, Nanjing University, Suzhou 215163, China. (E-mail: ywen@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).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI: 10.1109/TMC.2025.3582683, IEEE Transactions on Mobile Computing

challenging to precisely quantify how logits aggregation errors impact overall model performance.

- *Errors due to channel fading and noise:* AirComp is susceptible to errors introduced by channel attenuation and stochastic noise during logits aggregation. While FedAvg and FedSGD can mitigate these errors through effective transmission power control and denoising strategies, Air-CoKD faces heightened difficulty in reducing aggregation errors due to the dimensional inconsistencies in locally generated logits caused by unbalanced device data class in non-IID settings.

In this paper, we propose a logits aggregation mechanism using orthogonal frequency division multiplexing (OFDM) to address the dimensional inconsistencies caused by unbalanced device data class during FedKD combined with AirComp. This method assigns orthogonal subcarriers to logits of different target classes, allowing simultaneous transmission across multiple subcarriers from various devices. This approach achieves one-shot aggregation per training round but introduces the challenge of multi-user and multi-channel allocation. To resolve this, we analyze the convergence of the OFDM-based Air-CoKD and quantify the impact of aggregation errors on model performance. Given the complexity of the KL divergence in local loss functions, we approximate it with a quadratic upper bound to simplify analysis. We then formulate an optimization problem to minimize the convergence upper bound while considering constraints on bandwidth, transmission power, and device power budgets. This problem, being a mixed integer nonlinear programming (MINLP) problem, is tackled through alternating optimization, decoupling it into manageable convex and mixed-integer subproblems. Finally, extensive simulations on MNIST and CIFAR-10 datasets show that Air-CoKD can effectively balances model training efficiency and performance, making it well-suited for resource-constrained environments. By combining KD and AirComp techniques, our method can reduce communication overhead by 90% at per round and significantly improve transmission efficiency. The main contributions of this paper includes:

1) We introduce a novel CML framework, Air-CoKD, which integrates AirComp with FedKD to balance training delay and model performance in resource-constrained networks. By proposing an OFDM-based logits aggregation mechanism, it achieves one-shot aggregation per training round by assigning orthogonal subcarriers to logits of different target classes, effectively mitigating dimensional inconsistencies caused by unbalanced device data class.
2) We provide a convergence analysis for models trained with Air-CoKD, applicable to both convex and non-convex scenarios. This analysis quantifies the impact of logits aggregation errors on model performance, identifying the mean square error (MSE) of aggregated logits as a key factor in determining the convergence upper bound.
3) Based on the convergence analysis, an optimization problem is formulated and solved to minimize the MSE of aggregated logits. An alternating optimization algorithm is proposed to address this MINLP problem, delivering high-quality solutions with polynomial computational complexity.

The rest of the paper is organized as follows: Section II presents the preliminaries, Section III reviews related works, Section IV presents the system model, Section V details the convergence analysis, Section VI formulates the optimization problem, Section VII proposes the alternating optimization algorithm, Section VIII discusses the simulation results, and Section IX concludes the paper.

II. Related Works

CML's bandwidth and energy limitations in resource-constrained networks can be addressed with KD and AirComp. This section reviews FedKD and AirComp methods, focusing on optimizing model performance, communication efficiency, and resource allocation.

In FedKD, lightweight logits are transferred between devices and the parameter server (PS) to enhance communication efficiency. Chen et al. [15] optimized device scheduling and communication resource allocation using network knowledge instead of model parameters. Liu et al. [16] improved performance by uploading both logits and model parameters to the PS. Mishra et al. [17] grouped clients by bandwidth and used KD to compress model information. Zhu et al. [18] proposed a data-free FedKD method using a lightweight generator to replace proxy datasets. Deng et al. [19] introduced a multi-layer KD-based FL framework, clustering devices by data distribution. Existing FedKD solutions focus on algorithm design but often overlook transmit-receive strategies, making them less suitable for resource-sensitive networks.

AirComp-based CML methods use analog uplink transmission for parameter aggregation, improving efficiency but facing challenges in designing efficient transmit-receive strategies. Cao et al. [20], [21] analyzed the relationship between aggregation errors and model performance. Guo et al. [22] explored model convergence under uplink and downlink channel errors and designed joint optimization for device selection and power control. Du et al. [23] proposed dynamic device scheduling based on gradients, channel conditions, and energy constraints to enhance model training efficiency. To further improve efficiency, compression mechanisms such as gradient sparsity and 1-bit quantization have been explored in [24], [25]. Ahn et al. [26] ensured consistent local gradient dimensions by sharing a global sparsity pattern. Despite reducing overhead, selecting compression parameters remains challenging, and excessive compression can lead to information loss. Channel fading and noise can also slow convergence, increasing training delays.

III. Preliminaries

This section first reviews model parameter transfer and logits-based knowledge transfer methods in CML. It then introduces the OFDM-based AirComp approach, which addresses dimensional inconsistencies caused by unbalanced device data class.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI: 10.1109/TMC.2025.3582683, IEEE Transactions on Mobile Computing

A. Model Parameter Transfer Based CML

The most commonly used model parameter transfer based CML is the vanilla FL method, such as `FedAvg` or `FedSGD` [27]. We assume that a set $\mathcal{K}$ of K IoT devices and a PS in a CML system. Denote the dataset collected by any device k ($\forall k \in \mathcal{K}$) as $\mathcal{D}_k = \{(\mathbf{x}_{k,i}, y_{k,i})\}_{i=1}^{D_k}$, where $D_k = |\mathcal{D}_k|$ represents the size of $\mathcal{D}_k$, $\mathbf{x}_{k,i} \in \mathbb{R}^n$ denotes the i-th ($\forall i \in \{1, \cdots, D_k\}$) sample with dimension n, and $y_{k,i} \in \mathbb{R}$ represents the corresponding label. Let $\mathcal{D} = \bigcup_{k\in\mathcal{K}} \mathcal{D}_k$ denote the dataset aggregated across all K devices. The total size of $\mathcal{D}$ is given by $D = |\mathcal{D}| = \sum_{k=1}^{K} D_k$.

The PS first broadcasts the complete model parameter $\mathbf{w} \in \mathbb{R}^q$ (where q is the size of the parameter vector) to the devices for local training. Each device k uses its dataset $\mathcal{D}_k$ to train the model $\mathbf{w}$, and then uploads the trained model or gradient back to the PS for aggregation. Denote the local loss function of device k as

$$F_k(\mathbf{w}) = \frac{1}{D_k} \sum_{i=1}^{D_k} \mathcal{L}\left(\mathbf{w}, (\mathbf{x}_{k,i}, y_{k,i})\right), \ \forall k \in \mathcal{K}, \tag{1}$$

where $\mathcal{L}(\mathbf{w}, (\mathbf{x}_{k,i}, y_{k,i}))$ represents the empirical sample-wise loss determined by a specific learning task. This process is iteratively repeated to minimize the global loss $F(\mathbf{w})$ given below by optimizing the model parameter $\mathbf{w}$.

$$\min_{\mathbf{w}} F(\mathbf{w}) = \sum_{k\in\mathcal{K}} \frac{D_k}{D} F_k(\mathbf{w}). \tag{2}$$

Remark 1. *Parameter-sharing CML methods require devices to upload complete or partial model parameters at each round. For example, transferring data between a device and the PS for 50 training rounds using the lightweight model EfficientNetV2-S [28] involves 2GB of data. In post-disaster scenarios with limited communication bandwidth, this data transfer requirement significantly hampers training efficiency. Additionally, the energy consumption associated with such data transfers is unsustainable for battery-powered devices.*

B. Knowledge Transfer Based CML

Unlike model parameter aggregation, knowledge aggregation updates the model by sharing experience from devices rather than directly transferring model parameters. We employ the `FedKD` mechanism, where the aggregated model output across all devices serves as the teacher, and each device's local model is the student [29]. This approach significantly reduces communication traffic in both the uplink and downlink. Upon receiving the initial model $\mathbf{w}$ from the PS, each device k localizes it as model $\mathbf{w}_k \in \mathbb{R}^q$ and trains $\mathbf{w}_k$ using the local dataset $\mathcal{D}_k$. In addition to the basic loss F_k as indicated in eq. (1), `FedKD` incorporates a KD regularizer based on KL divergence to guide local training:

$$F_{\mathrm{KD}} = \frac{1}{D_k} \sum_{i=1}^{D_k} \mathbf{KL}\left(\widetilde{\phi}(y_{k,i}) \,||\, \phi_k(\mathbf{w}_k, \mathbf{x}_{k,i})\right), \ \forall k \in \mathcal{K}, \tag{4}$$

where $\phi_k(\mathbf{w}_k, \mathbf{x}_{k,i})$ represents the local logits inferred by $\mathbf{w}_k$ from $\mathbf{x}_{k,i}$, and $\widetilde{\phi}(y_{k,i})$ represents the global logits corresponding to the label $y_{k,i}$. Let $\gamma_k > 0$ denote the regularization parameter. The local loss function $Q_k(\mathbf{w}_k)$ for device k is given by eq. (3).

Denote $\mathcal{M} = \{1, 2, \ldots, M\}$ as the set of classes across all local datasets $\mathcal{D}_k, \forall k \in \mathcal{K}$, with M representing the total number of classes. The global logits $\widetilde{\phi}(\cdot)$ have a dimension of $\mathbb{R}^{M\times M}$. Let $\phi_{k,m}(\mathbf{w}_k) \in \mathbb{R}^M$ denote the vector of local logits for the m-th class label, with each element $[\phi_{k,m}(\mathbf{w}_k)]_j$ representing the probability for the j-th class label. These logits satisfy[1]

$$[\phi_{k,m}(\mathbf{w}_k)]_j \geq 0, \ \forall j \in \mathcal{M}, \ \sum_{\forall j\in\mathcal{M}} [\phi_{k,m}(\mathbf{w}_k)]_j = 1. \tag{5}$$

The PS then aggregates the local logits from all devices by computing a label-wise average to obtain the global logits. Under the assumption of an ideal data distribution, where each device has all M-class data, the aggregation operation for the m-th class is given by

$$\widetilde{\phi}_m = \frac{1}{K} \sum_{i=1}^{K} \phi_{k,m}, \ \forall m \in \mathcal{M}. \tag{6}$$

These global logits represent the collective knowledge learned through the KD mechanism and are fed back to the devices for the next training round, as shown in eq. (3).

Remark 2. *While `FedKD` only requires devices to upload local logits for aggregation after each training round, the limited bandwidth of wireless channels poses a challenge for accommodating such large-scale concurrent communication.*

C. OFDM-based Over-the-Air Computation

By utilizing the analog signal superposition mechanism in the Multiple Access Channel (MAC), AirComp efficiently extracts functions like arithmetic averages, enabling large-scale concurrent logit communication. However, device data imbalance leads to dimensional inconsistencies in local logits, hindering seamless aggregation. With reference to [30], [31], [32], we propose a logits aggregation mechanism based on OFDM and AirComp, where each subcarrier is assigned to

[1]The local model $\mathbf{w}_k$ maintains the same architecture with the global model $\mathbf{w}$. Consequently, the output dimension of $\mathbf{w}$ is determined by the number of global labels M, which means the size of the logits vector $\phi_{k,m}(\mathbf{w}_k)$ is M.

$$\begin{aligned} Q_k(\mathbf{w}_k) &= F_k(\mathbf{w}_k) + \frac{\gamma_k}{2} F_{\mathrm{KD}}(\mathbf{w}_k) \\ &= \frac{1}{D_k} \sum_{i=1}^{D_k} \mathcal{L}_k(\mathbf{w}_k, (\mathbf{x}_{k,i}, y_{k,i})) + \frac{\gamma_k}{2D_k} \sum_{i=1}^{D_k} \mathbf{KL}\left(\widetilde{\phi}(y_{k,i}) \,||\, \phi_k(\mathbf{w}_k, \mathbf{x}_{k,i})\right), \ \forall k \in \mathcal{K}. \end{aligned} \tag{3}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI: 10.1109/TMC.2025.3582683, IEEE Transactions on Mobile Computing

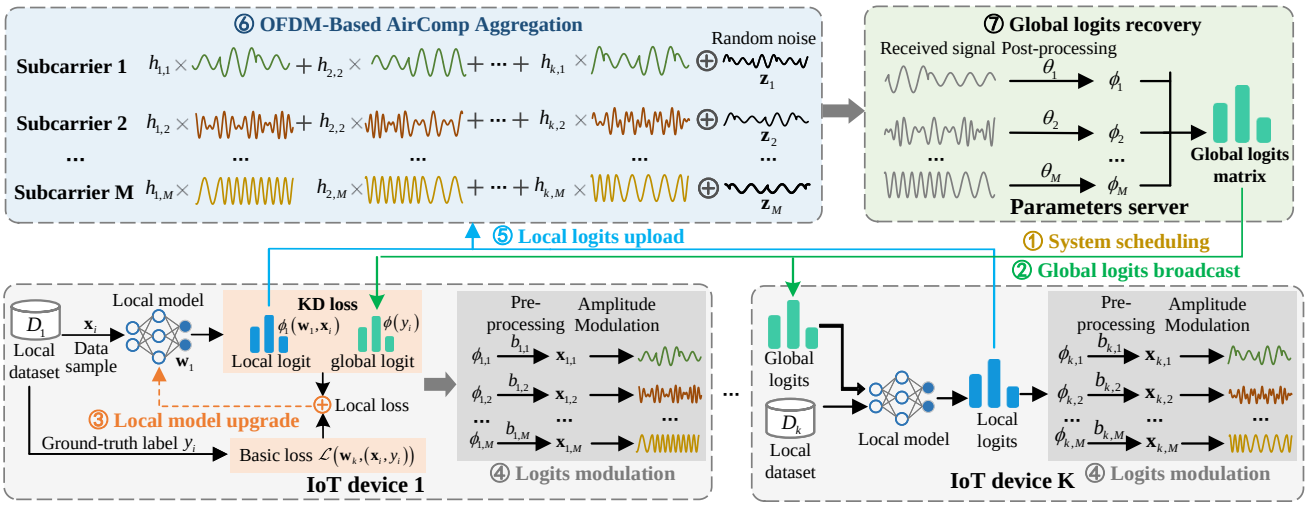


Fig. 1. The system model and workflow of our Air-CoKD framework.

the logits of a specific class. This design enables one-shot aggregation per training round and mitigates dimensional mismatches caused by class imbalance. Furthermore, our approach supports subcarrier-level optimization, offering fine-grained control over logits aggregation performance and more efficient spectrum utilization than existing methods.

Assuming a quasi-static Rayleigh fading channel, where the channel coefficient remains constant within each training round but can vary independently from one round to the next, the complex channel coefficient of device k on subcarrier $m \in \mathcal{M}$ in round t is represented as $h_{k,m}^t \in \mathbb{C}$. Thus, in the communication round t, the received signal on subcarrier m at the PS is represented by

$$\mathbf{y}_m^t = \sum_{k\in\mathcal{K}} h_{k,m}^t \mathbf{x}_{k,m}^t + \mathbf{z}_m^t, \ \forall m \in \mathcal{M}, \tag{7}$$

where $\mathbf{x}_{k,m}^t$ is the transmitted signal on on subcarrier m of any device k and $\mathbf{z}_m^t \sim \mathcal{CN}\left(0, \sigma^2\right) \in \mathbb{R}^M$ represents the additive Gaussian white noise (AWGN).

IV. System Model

After a disaster, AI models can predict its progression, aiding trapped individuals and rescue teams in avoiding danger and executing effective strategies. To enhance the real-time performance and accuracy of prediction, timely updates to these AI models are essential. Assume K IoT devices remain operational in the disaster zone, including portable terminals carried by individuals and environmental monitoring devices like cameras and sensors. With traditional communication and computing infrastructure disabled, a rotary drone equipped with communication and computing capabilities serves as the PS, establishing direct wireless links with all devices in set $\mathcal{K}$ and coordinating distributed training of AI models via CML.

This paper integrates AirComp with `FedKD` to enhance bandwidth and energy efficiency and improve training performance in resource-constrained networks. A major challenge arises from inconsistent dimensions of local logits across devices, hindering AirComp's signal superposition. To resolve this, a local logits alignment scheme based on OFDM is introduced, forming the `Air-CoKD` framework, as illustrated in Fig. 1. The following sections describe two critical steps: *devices training local models to generate logits* and *the PS aggregating these logits*. Subsequently, the complete workflow of the framework is presented.

A. Local Model Upgrade and Logits Generation

The proposed `Air-CoKD` framework allows each device k to update its local model $\mathbf{w}_k$ using stochastic gradient descent (SGD). After each t-th $(t = 1, 2, \cdots)$ round of training, the local model $\mathbf{w}_k^t$ of device k is updated as follows

$$\mathbf{w}_k^{t+1} = \mathbf{w}_k^t - \eta^t \nabla Q_k\left(\mathbf{w}_k^t, \mathcal{B}_k^t\right), \ \forall k \in \mathcal{K}, \tag{8}$$

where η^t denotes the learning rate at round t, $\mathcal{B}_k^t$ is the mini-batch sampled randomly from the local dataset $\mathcal{D}_k$ with size $|\mathcal{B}_k^t| = B$, and $\nabla Q_k\left(\mathbf{w}_k^t, \mathcal{B}_k^t\right)$ is the local stochastic gradient. The gradient is computed as

$$\begin{aligned}\nabla Q_k\left(\mathbf{w}_k^t, \mathcal{B}_k^t\right) &= \frac{1}{B}\sum_{\xi\in\mathcal{B}_k^t} \nabla Q_k\left(\mathbf{w}_k^t, \xi\right) \\ &= \frac{1}{B}\sum_{\xi\in\mathcal{B}_k^t} \nabla F_k\left(\mathbf{w}_k^t, \xi\right) + \frac{\gamma_k}{2}\nabla F_{\mathrm{KD}}\left(\mathbf{w}_k^t, \xi\right), \ \forall k \in \mathcal{K},\end{aligned} \tag{9}$$

where ξ denotes any i-th pair $(\mathbf{x}_{k,i}, y_{k,i})$ of the mini-batch $\mathcal{B}_k^t$, and $\nabla F_k(\cdot)$ and $\nabla F_{\mathrm{KD}}(\cdot)$ represent the stochastic gradients of the basic loss F_k and the KD loss F_{KD}, respectively.

During training, each device generates local logits with local training data. For a subset of $\mathcal{B}_{k,m}^t \subset \mathcal{D}_k$ with class label m and size $B_{k,m}^t$, the local logits are averaged as

$$\phi_{k,m}\left(\mathbf{w}_k^t\right) = \frac{1}{B_{k,m}^t}\sum_{i=1}^{B_{k,m}^t} \phi_k\left(\mathbf{w}_k^t, \mathbf{x}_{k,i}\right), \ \forall k \in \mathcal{K}, \ \forall m \in \mathcal{M}, \tag{10}$$

where $\phi_k\left(\mathbf{w}_k^t, \mathbf{x}_{k,i}\right)$ represents the sample-wise logits.

In non-IID settings, devices may not have identical data classes. Let $\mathcal{M}_k = \{1, \cdots, M_k\} \subseteq \mathcal{M}$ denote the classes

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI: 10.1109/TMC.2025.3582683, IEEE Transactions on Mobile Computing

present in dataset $\mathcal{D}_k$, with size M_k. The device generates a local logits matrix

$$\phi_k^t = \left[\phi_{k,1}\left(\mathbf{w}_k^t\right); \cdots; \phi_{k,M_k}\left(\mathbf{w}_k^t\right)\right] \in \mathbb{R}^{M_k \times M},\ \forall k \in \mathcal{K}, \tag{11}$$

which is then uploaded to the PS for aggregation. To ensure comprehensive knowledge acquisition, the dataset $\mathcal{B}_k^t$ must cover all classes in $\mathcal{M}_k$.

B. OFDM-based Logits Aggregation

A key step in each training round is using AirComp to aggregate local logits uploaded by devices to form the global logits. Due to the unbalanced device data class M_k, the dimensions of the local logits matrix $\phi_k^t \in \mathbb{R}^{M_k \times M}$ for each device k varies, which complicates AirComp's superposition requirements and necessitates additional label-wise aggregation rounds.

To address this, an OFDM-based method is proposed to align the dimension of local logits. The available bandwidth is divided into M orthogonal subcarriers, with each subcarrier corresponding to one of the M classes in the global dataset $\mathcal{D} = \bigcup_{k\in\mathcal{K}} \mathcal{D}_k$. The m-th subcarrier is exclusively assigned for uploading logits associated with the m-th class, ensuring dimension alignment across all devices. This setup introduces a subcarrier allocation problem, defined by $a_{k,m}^t \in \{0,1\}$, where $a_{k,m}^t = 1$ indicates that device k is allocated subcarrier m for uploading logits in the t-th round of training, and $a_{k,m}^t = 0$ otherwise.

Remark 3. *Unlike traditional subcarrier allocation problems in OFDM systems,* `Air-CoKD` *permits each device k to utilize multiple subcarriers to transmit logits of different classes of data. Consequently, each subcarrier m carries logits of the same data class sent in parallel by multiple devices. Thus, the dimension of the subcarrier allocation matrix $\boldsymbol{A} = [a_{k,m}] \in \mathbb{R}^{K\times M}$ is $K \times M$.*

Assuming perfect knowledge of the CSI between the PS and devices, along with the subcarrier and power allocation scheme from the PS, the transmitted signal from any device k is designed as

$$\mathbf{x}_{k,m}^t = b_{k,m}^t \phi_{k,m}^{t-1}, \tag{12}$$

where $b_{k,m}^t = a_{k,m}^t \frac{\sqrt{p_{k,m}^t}\left(h_{k,m}^t\right)^{\mathrm{H}}}{\left|h_{k,m}^t\right|}$ denotes the pre-processing coefficient of device k at round t. Then, the signal received by the PS on subcarrier m is given by

$$\mathbf{y}_m^t = \sum_{k\in\mathcal{K}} a_{k,m}^t \sqrt{p_{k,m}^t}\left|h_{k,m}^t\right| \phi_{k,m}^{t-1} + \mathbf{z}_m^t,\ \forall m \in \mathcal{M}, \tag{13}$$

where $p_{k,m}^t \geq 0$ denotes the transmission power scalar of device k on subcarrier m in round t. Considering the maximum allowable power budget $P_{k,m}^{\max}$ of device k on subcarrier m, and the total power budget P_k^{tol} of device k in round t, the following constraints must be satisfied:

$$\left\| a_{k,m}^t \sqrt{p_{k,m}^t} \phi_{k,m}^{t-1} \right\|^2 \leq P_k^{\max},\ \forall k \in \mathcal{K}, \tag{14}$$

and

$$\sum_{m=1}^{M} \left\| a_{k,m}^t \sqrt{p_{k,m}^t} \phi_{k,m}^{t-1} \right\|^2 \leq P_k^{\mathrm{tol}},\ \forall k \in \mathcal{K}. \tag{15}$$

Let $\mathcal{S}_m^t$ denote the set of devices selected in round t to transmit logits on subcarrier m. The size of $\mathcal{S}_m^t$ is $|S_m^t| = \sum_{k=1}^K a_{k,m}^t$. Thus, the global logits vector of class m outputted by AirComp is

$$\begin{aligned}\phi_m^t &= \frac{\mathbf{y}_m^t}{\sqrt{\theta_m^t}\,|S_m^t|} \\ &= \frac{\sum_{k\in\mathcal{K}} a_{k,m}^t \sqrt{p_{k,m}^t}\left|h_{k,m}^t\right|}{\sqrt{\theta_m^t}\,|S_m^t|} \phi_{k,m}^{t-1} + \frac{\mathbf{z}_m^t}{\sqrt{\theta_m^t}\,|S_m^t|},\ \forall m \in \mathcal{M},\end{aligned} \tag{16}$$

where θ_m^t is the denoising factor on subcarrier m in round t.

Considering a more general case that encompasses both balanced and unbalanced device data distribution settings, the error-free logits aggregation on subcarrier m is updated from eq. (6) to the following form:

$$\widetilde{\phi}_m^t = \frac{1}{|S_m^t|} \sum_{i=1}^{K} a_{k,m}^t \phi_{k,m}^{t-1},\ \forall m \in \mathcal{M}. \tag{17}$$

Thus, the aggregation error of the global logits output by AirComp is

$$\mathbf{e}_m^t = \phi_m^t - \widetilde{\phi}_m^t,\ \forall m \in \mathcal{M}. \tag{18}$$

C. The Complete Workflow of `Air-CoKD`

Fig. 1, along with the two key steps and related data formats, illustrates the complete workflow of the proposed `Air-CoKD` as follows:

1) System Scheduling: At the beginning of each communication round t, the PS transmits OFDM subcarrier and power allocation strategies to the devices. Based on this, each device modulates its local logits into analog signals and allocates the transmitted signals across different subcarriers.

2) Model and Logits Broadcasting: At the initial stage ($t = 0$), the PS broadcasts the global model $\mathbf{w}$ to all devices to initialize their local models as $\mathbf{w}_k = \mathbf{w}, \forall k \in \mathcal{K}$. In subsequent training rounds ($t \geq 1$), the PS broadcasts only the aggregated global logits matrix to all devices.

3) Local Model Update: Upon receiving the global logits, each device k trains its local model by minimizing a combination of KD loss and basic loss as described in Section IV-A, generating local logits for the next communication round.

4) Logits Modulation: Each device modulates its local logits into analog signals according to the scheduling strategy, applies a pre-processing coefficient to compensate for channel phase, and adds a cyclic prefix (CP) to the OFDM symbols to avoid inter-symbol interference (ISI).

5) Local Logits Upload: After local training and modulation, each device transmits the modulated logits signals on the assigned subcarriers. To ensure time synchronization, devices estimate propagation delay by measuring the time difference between pilot signal transmission and reception, then transmit the OFDM symbols in advance.

6) OFDM-Based AirComp Aggregation: The logits analog signals from all devices are superposed across subcarriers.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI: 10.1109/TMC.2025.3582683, IEEE Transactions on Mobile Computing

Channel attenuation and additive noise introduce aggregation errors in the global logits signal, as discussed in Section IV-B.

7) Global Logits Recovery: The PS applies denoising and averaging operations to the received signal to reconstruct the aggregated global logits.

V. Convergence analysis of Air-CoKD

The designed Air-CoKD improves resource utilization in CML. However, the aggregation errors in global logits introduced by AirComp, as given in eq. (18), can potentially degrade model performance. This section quantifies the impact of these aggregation errors on model effectiveness, which is crucial for error mitigation and enhancing model performance.

A. Upper Bound of Local Loss

The convergence of Air-CoKD is initially influenced by the local loss function $Q_k\left(\mathbf{w}_k^t\right)$, as shown in eq. (3). The KD loss component $F_{\mathrm{KD}}\left(\cdot\right)$ uses KL divergence to measure the discrepancy between global and local logits. However, the absence of a closed-form derivative for KL divergence complicates the analysis of the error term's impact on distillation loss. To address this, we use the inequality properties of KL divergence to derive a quadratic upper bound for $Q_k\left(\mathbf{w}_k^t\right)$, following the approach in [33].

Let $\phi^t\left(y_{k,i}\right)$ represent the global logits for the class label $y_{k,i}$ at round t, computed by aggregating the local logits from selected devices in round $t-1$ via AirComp. Define $\phi_k\left(\mathbf{w}_k^t, \mathbf{x}_{k,i}\right) \in \mathbb{R}^M$ as the vector of local logits derived from $\mathbf{x}_{k,i}$ using $\mathbf{w}_k^t$. The upper bound on $Q_k\left(\mathbf{w}_k^t\right)$ is given in the following lemma.

Lemma 1. *Let $\delta > 0$ be a positive constant such that $\min_{j\in\mathcal{M}}\left[\phi_k\left(\mathbf{w}_k, \mathbf{x}_{k,i}\right)\right]_j \geq \delta$. The local loss $Q_k\left(\mathbf{w}_k^t\right)$ for any device k is bounded by*

$$Q_k\left(\mathbf{w}_k^t\right) \leq \widetilde{Q}_k\left(\mathbf{w}_k^t\right) = F_k\left(\mathbf{w}_k^t\right) + \frac{\gamma_k}{2\delta}\widetilde{F}_{\mathrm{KD}}\left(\mathbf{w}_k^t\right), \quad (19)$$

where

$$\widetilde{F}_{\mathrm{KD}}\left(\mathbf{w}_k^t\right) = \frac{1}{D_k}\sum_{i=1}^{D_k}\left\|\phi^t\left(y_{k,i}\right) - \phi_k\left(\mathbf{w}_k^t, \mathbf{x}_{k,i}\right)\right\|^2. \quad (20)$$

Proof. For a detailed proof, please refer to Appendix A. □

According to **Lemma 1**, when $Q_k\left(\mathbf{w}_k^t\right)$ and $\widetilde{Q}_k\left(\mathbf{w}_k^t\right)$ achieve their respective optima at $\mathbf{w}_k^*$ and $\widetilde{\mathbf{w}}_k^*$, it follows that $Q_k(\mathbf{w}_k^*) \leq Q_k(\widetilde{\mathbf{w}}_k^*) \leq \widetilde{Q}_k(\widetilde{\mathbf{w}}_k^*)$. This indicates that as $Q_k\left(\mathbf{w}_k^t\right)$ approaches its optimum, $\widetilde{Q}_k\left(\mathbf{w}_k^t\right)$ converges to its optimal value. Given the presence of aggregation errors, the optimal local loss $Q_k\left(\mathbf{w}_k^t\right)$ can be approximated by minimizing the upper bound $\widetilde{Q}_k\left(\mathbf{w}_k^t\right)$. Thus, $\widetilde{Q}_k\left(\mathbf{w}_k^t\right)$ is used as the local loss function for the subsequent convergence analysis.

B. Convergence Analysis

As outlined in eq. (3), the local model $\mathbf{w}_k$ is trained using the KD loss $\widetilde{F}_{\mathrm{KD}}\left(\mathbf{w}_k\right)$, while the convergence of the model is governed by the base loss function $F_k\left(\mathbf{w}_k\right)$. We adopt the following assumptions for analysis, consistent with [21], [34].

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

$$\left\|\nabla F_k\left(\mathbf{w}\right) - \nabla F_k\left(\mathbf{v}\right)\right\| \leq L\left\|\mathbf{w} - \mathbf{v}\right\|, \quad (21)$$

which extends to

$$F_k\left(\mathbf{w}\right) \leq F_k\left(\mathbf{v}\right) + \left\langle\nabla F_k\left(\mathbf{w}\right), \mathbf{w} - \mathbf{v}\right\rangle + \frac{L}{2}\left\|\mathbf{w} - \mathbf{v}\right\|^2. \quad (22)$$

Assumption 2. *(Bounded Stochastic Gradient Variance) The local stochastic gradients are unbiased, i.e.,*

$$\mathbb{E}\left[\nabla F_k\left(\mathbf{w}, \mathbf{x}_{k,i}\right)\right] = \nabla F_k\left(\mathbf{w}\right), \quad (23)$$

and the variance of stochastic sample gradient $\nabla F_k\left(\mathbf{w}, \mathbf{x}_{k,i}\right)$ is bounded by a positive constant μ_k, i.e.,

$$\mathbb{E}\left[\left\|\nabla F_k\left(\mathbf{w}, \mathbf{x}_{k,i}\right) - \nabla F_k\left(\mathbf{w}\right)\right\|^2\right] \leq \mu_k^2. \quad (24)$$

Assumption 3. *(Gradient Bound) The squared norm of stochastic sample gradient $\left\|\nabla\phi_k\left(\mathbf{w}, \mathbf{x}_{k,i}\right)\right\|^2$ is bounded by a positive constant G, i.e.,*

$$\left\|\nabla\phi_k\left(\mathbf{w}, \mathbf{x}_{k,i}\right)\right\|^2 \leq G^2, \quad (25)$$

where $\nabla\phi_k\left(\mathbf{w}, \mathbf{x}_{k,i}\right)$ is the gradient of the model output (logits), often used to construct the neural tangent kernel (NTK) [35].

We next investigate the impact of logits aggregation errors on model performance under both convex and non-convex scenarios.

1) Convex case: We assume the base loss function $F_k\left(\cdot\right)$ exhibits strong convexity, as defined below.

Assumption 4. *(Strong Convexity) For any model parameter $\mathbf{w}, \mathbf{v} \in \mathbb{R}^q$, the gradient of $F_k\left(\cdot\right)$, represented by $\nabla F_k\left(\mathbf{w}\right)$, is strong convexity with constant $\rho > 0$, meaning that*

$$F_k\left(\mathbf{w}\right) \geq F_k\left(\mathbf{v}\right) + \left\langle\nabla F_k\left(\mathbf{w}\right), \mathbf{w} - \mathbf{v}\right\rangle + \frac{\rho}{2}\left\|\mathbf{w} - \mathbf{v}\right\|^2, \ \forall k \in \mathcal{K}. \quad (27)$$

Under this assumption, F_k has a unique optimal value F_k^*. The following theorem provides the bound on the gap between the expected loss $\mathbb{E}\left[F_k\left(\mathbf{w}_k^{t+1}\right)\right]$ and the optimal value F_k^*.

Theorem 1. *Given an initial local model $\mathbf{w}_k^0$ and a learning rate $0 < \eta^t \leq \frac{1}{2L}$, after T rounds of training, the bound on $\mathbb{E}\left[F_k\left(\mathbf{w}_k^{t+1}\right) - F_k^*\right]$ is given by*

$$\begin{aligned}
&\mathbb{E}\left[F_k\left(\mathbf{w}_k^{t+1}\right) - F_k^*\right] \\
&\leq \left(1 - \rho\eta_t\left(\frac{1}{2} - L\eta_t\right)\right)\mathbb{E}\left[F_k\left(\mathbf{w}_k^t\right) - F_k^*\right] \\
&+ 4\eta_t\left(L\eta_t + \frac{1}{2}\right)\frac{\gamma_k^2 M_k G^2}{\delta^2}\sum_{m=1}^{M_k}\Omega_m^t + L\eta_t^2\frac{\mu_k^2}{B} \\
&+ 8\eta_t\left(L\eta_t + \frac{1}{2}\right)\frac{\gamma_k^2 M_k G^2}{\delta^2} \\
&= A_k^t\mathbb{E}\left[F_k\left(\mathbf{w}_k^t\right) - F_k^*\right] + C_k^t,
\end{aligned} \quad (28)$$

where

$$A_k^t = 1 - \rho\eta_t\left(\frac{1}{2} - L\eta_t\right), \quad (29)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI: 10.1109/TMC.2025.3582683, IEEE Transactions on Mobile Computing

and

$$C_k^t = 4\eta_t\left(L\eta_t+\frac{1}{2}\right)\frac{\gamma_k^2 M_k G^2}{\delta^2}\sum_{m=1}^{M_k}\Omega_m^t + L\eta_t^2\frac{\mu_k^2}{B} + 8\eta_t\left(L\eta_t+\frac{1}{2}\right)\frac{\gamma_k^2 M_k G^2}{\delta^2}. \tag{30}$$

with

$$\Omega_m^t = \sum_{k\in\mathcal{K}}\frac{a_{k,m}^t}{|S_m^t|^2}\left(\frac{\sqrt{p_{k,m}^t}|h_{k,m}^t|}{\sqrt{\theta_m^t}}-1\right)^2 + \frac{\sigma^2}{\theta_m^t|S_m^t|^2}. \tag{31}$$

Proof. For a detailed proof, please refer to Appendix B. □

From eq. (28), for $0 < \eta^t \leq \frac{1}{2L}$, $A_k^t \leq 1$ and $\lim_{T\to\infty}\prod_{t=1}^{T} A_k^t = 0$. Consequently, the first term on the right-hand side of eq. (28) approaches 0, and the gap is primarily determined by C_k^t. Eq. (30) shows that C_k^t consists of three terms: Term (a) represents the logits aggregation error introduced by the wireless channel, where Ω_m^t in eq. (31) denotes the MSE on subcarrier m at round t. Term (b) reflects the variance of the stochastic gradient associated with the base loss F_k. Term (c) captures the knowledge gap between the global and local logits at round t.

Generally, terms (b) and (c) depend on data distribution and learning strategy, and are not directly affected by aggregation errors. For instance, increasing the local mini-batch size can help control term (b) and reduce the gap. Regarding term (a), minimizing channel interference from AirComp can bring Ω_m^t close to zero, achieving perfect aggregation of local logits. Meanwhile, the noise introduced by AirComp-based aggregation influences the model convergence performance by increasing the logits aggregation error Ω_m^t. A higher noise variance σ^2 leads to a larger Ω_m^t, which subsequently results in a looser (i.e., larger) upper bound on the model convergence. Thus, optimizing subcarrier allocation $\mathbf{A} = [a_{k,m}] \in \mathbb{R}^{K\times M}$, transmission power control $\mathbf{P} = [p_{k,m}] \in \mathbb{R}^{K\times M}$, and denoising factor $\Theta = (\theta_1, \cdots, \theta_M)$ can reduce the upper bound of the gap and accelerate convergence.

2) Non-convex case: When **Assumption 4** does not hold, indicating that $F_k(\cdot)$ is non-convex, the expected squared norm of the gradient functions as an indicator of model convergence. The global loss function is considered to achieve a ϵ-suboptimal solution if $\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla F\left(\mathbf{w}^t\right)\right\|^2\right] \leq \epsilon$, which guarantees that the training model converges to a stationary point [36]. The following theorem provides the conditions necessary for this convergence.

Theorem 2. *Given a learning rate $0 < \eta_t \leq \frac{1}{2L}$, after T rounds of training, the bound on the expected square norm of the gradient $\mathbb{E}\left[\left\|\nabla F\left(\mathbf{w}_k^t\right)\right\|^2\right]$ is given by eq. (26).*

Proof. For a detailed proof, please refer to Appendix C. □

Theorem 2 shows that for a non-convex $F_k(\cdot)$, the upper bound on $\mathbb{E}\left[\left\|\nabla F\left(\mathbf{w}_k^t\right)\right\|^2\right]$ can be reduced by minimizing the aggregation error C_k^t. Consequently, reducing this aggregation error also decreases the upper bound on the gap between the expected loss $\mathbb{E}\left[F_k\left(\mathbf{w}_k^{t+1}\right)\right]$ and the optimal value F_k^*.

VI. Problem Formulation

The cumulative error of global logits caused by AirComp over multiple rounds of logits aggregation significantly affects the convergence of `Air-CoKD`. According to **Theorem 1** and **Theorem 2**, whether the basic loss function $F_k(\cdot)$ is convex or non-convex, the convergence upper bound is primarily determined by C_k^t, which is influenced by the MSE Ω_m^t given in eq. (31). We denote the aggregation error of device k in round t as

$$\Psi_k^t = \sum_{\forall m\in\mathcal{M}_k} a_{k,m}^t\Omega_m^t,\ \forall k\in\mathcal{K}. \tag{32}$$

To enhance the convergence of `Air-CoKD`, it is crucial to minimize the aggregation error Ψ_k^t for each device k in any training round t.

By omitting the index t, the optimization problem can be formulated as follows:

$$\mathcal{P}\mathbf{0}: \min_{\mathbf{A},\mathbf{P},\Theta}\sum_{\forall k\in\mathcal{K}}\left(\Psi_k = \sum_{\forall m\in\mathcal{M}_k} a_{k,m}\Omega_m\right), \tag{33a}$$

$$\text{s.t.}\quad 0 \leq p_{k,m} \leq P_{k,m}^{\max},\ \forall k\in\mathcal{K},\ \forall m\in\mathcal{M}, \tag{33b}$$

$$0 \leq \sum_{m=1}^{M} a_{k,m}p_{k,m} \leq P_k^{\text{tol}},\ \forall k\in\mathcal{K}, \tag{33c}$$

$$a_{k,m}\in\{0,1\},\ \forall k\in\mathcal{K},\ \forall m\in\mathcal{M}, \tag{33d}$$

$$\theta_m \geq 0,\ \forall m\in\mathcal{M}, \tag{33e}$$

where $\mathbf{A} = [a_{k,m}] \in \mathbb{R}^{K\times M}$ represents the subcarrier allocation matrix, $\mathbf{P} = [p_{k,m}] \in \mathbb{R}^{K\times M}$ represents the power allocation matrix, and $\Theta = (\theta_1, \cdots, \theta_M)$ represents the vector of denoising factors. Constraints (33b) and (33c) come from eqs. (14) and (15), considering the relationship that $\|\phi_{k,m}^{t-1}\| \leq 1$ according to eq. (5). These constraints represent the maximum allowable power and the total power budget for each device k, respectively. Constraint (33d) is an indicator for subcarrier allocation, as detailed in *Remark 3*, and constraint (33e) defines the permissible range for the denoising factor θ_m applied to subcarrier m.

To simplify the problem, we use the inequality

$$\sum_{k=1}^{K}\sum_{m=1}^{M_k}\mathbb{I}\{a_{k,m}=1\}\Omega_m \leq K\sum_{m=1}^{M}\Omega_m, \tag{34}$$

transforming problem $\mathcal{P}0$ into

$$\mathcal{P}1: \min_{\mathbf{A},\mathbf{P},\Theta}\sum_{m=1}^{M}\Omega_m, \quad \text{s.t.}\quad (33b)\sim(33e). \tag{35}$$

$$\min_{t\in\mathcal{T}}\mathbb{E}\left[\left\|\nabla F\left(\mathbf{w}_k^t\right)\right\|^2\right] \leq \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla F\left(\mathbf{w}_k^t\right)\right\|^2\right] \leq \frac{4L}{T}\mathbb{E}\left[F_k\left(\mathbf{w}_k^0\right)-F_k^*\right] + \frac{4L}{T}\sum_{t=1}^{T}C_k^t \tag{26}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI: 10.1109/TMC.2025.3582683, IEEE Transactions on Mobile Computing

Since problem $\mathcal{P}1$ involves both continuous variables $\mathbf{P}$ and Θ and discrete variable $\mathbf{A}$, it constitutes a mixed integer nonlinear programming (MINLP) problem with highly coupled system variables. In the following, we design a low-complexity algorithm to obtain a high-quality solution to problem $\mathcal{P}1$.

VII. Performance Optimization

In this section, we propose an alternative optimization-based algorithm to solve problem $\mathcal{P}1$. The core idea is to decouple the parameters to be optimized and then solve the resulting subproblems iteratively. By proving the convergence of this approach, we can obtain an approximate optimal solution to problem $\mathcal{P}1$.

A. Subcarrier Allocation based on Divide and Conquer

Given the transmission power allocation $\mathbf{P}$ and the denoising factor vector Θ, problem $\mathcal{P}1$ is simplified to

$$\mathcal{P}1.1: \quad \min_{\mathbf{A}} \sum_{m=1}^{M} \Omega_m, \quad \text{s.t.} \quad (33b). \tag{36}$$

Let $\mathbf{a}_m = [a_{1,m}, \cdots, a_{K,m}]^{\mathrm{T}} \in \mathbb{R}^K, \forall m \in \mathcal{M}$ denotes the column vectors of the subcarrier allocation matrix $\mathbf{A}$. Since $\mathbf{a}_m$ is independent of each others, problem $\mathcal{P}1.1$ can be transformed into M independent subproblems, each corresponding to a specific subcarrier m. The m-th subproblem is

$$\min_{\mathbf{a}_m} \Omega_m = \sum_{k=1}^{K} \frac{a_{k,m}}{|S_m|^2} \left(\frac{\sqrt{p_{k,m}}|h_{k,m}|}{\sqrt{\theta_m}} - 1 \right)^2 + \frac{\sigma^2}{\theta_m |S_m|^2}, \tag{37a}$$

$$\text{s.t.} \quad a_{k,m} \in \{0,1\}, \ \forall k \in \mathcal{K}. \tag{37b}$$

Problem (37) is an integer programming problem for which no standard solution methods are available. To address this within distributed training scenarios, we propose divide and conquer strategy [37], which allows each device k to optimize the subcarrier allocation independently. Specially, the algorithm initializes subcarrier allocation factor $a_{k,m}$ of device k on each subcarrier m to 0. Multiple values of Ω_m can then be computed. If the minimum Ω_m is less than the previously obtained value, the corresponding $a_{k,m}$ is retained. After multiple iterations, the objective value of Ω_m stablizes, leading to the optimal subcarrier allocation $\mathbf{A}$. The algorithm is described as follows.

In lines 2-3 of **Algorithm 1**, $\bar{\Omega}_m$ represents the dynamic lower bound of Ω_m, computed with the current subcarrier allocation $\mathbf{a}_m$. In line 5, each non-zero element in $\mathbf{a}_m$ is temporarily set to 0, and Ω_m is recalculated, generating a set Ξ of possible Ω_m values. In lines 7-8, the minimum Ω_m in set Ξ is identified as $\Omega_m^{\min} = \min(\Xi)$, and the corresponding device is marked as k^*. In lines 9-10, if $\Omega_m^{\min}$ is less than $\bar{\Omega}_m$, then $a_{k^*,m} = 0$, indicating that device k^* should not transmit on subcarrier m. This process is repeated until Ω_m stabilizes, yielding the optimal subcarrier allocation $\mathbf{a}_m$.

Algorithm 1: Subcarrier allocation algorithm.

Input: θ_m, $\{p_{k,m}\}_{k=1}^K$, $\{h_{k,m}\}_{k=1}^K$
Output: $\mathbf{a}_m$
1 Initialize: Subcarrier allocation vector $\mathbf{a}_m$, auxiliary variables $\Omega_m^{\min} = 0$ and $\bar{\Omega}_m = 0$, and list $\Xi = \emptyset$;
while $\Omega_m^{\min} < \bar{\Omega}_m$ **do**
2 Obtain Ω_m by substituting current $\mathbf{a}_m$ into eq. (37a);
3 $\bar{\Omega}_m \leftarrow \Omega_m$;
4 **for** $k \in \{1, 2, \ldots, K\}$ *and* $\mathbf{a}_m[k] \neq 0$ **do**
5 $\mathbf{a}_m[k] \leftarrow 0$. Obtain Ω_m by substituting current $\mathbf{a}_m$ into eq. (37a), and append it to Ξ;
6 $\mathbf{a}_m[k] \leftarrow 1$;
7 $\Omega_m^{\min} \leftarrow \min(\Xi)$;
8 Find device k^* that causes $\Omega_m^{\min}$;
9 **if** $\Omega_m^{\min} < \bar{\Omega}_m$ **then**
10 Update $\mathbf{a}_m$ by setting $\mathbf{a}_m[k^*] = 0$;
11 Clear list Ξ;

B. Denoising Factor Optimization

Given the subcarrier allocation $\mathbf{A}$ and power allocation $\mathbf{P}$, problem $\mathcal{P}1$ can be simplified to

$$\mathcal{P}1.2: \quad \min_{\Theta} \sum_{m=1}^{M} \Omega_m, \quad \text{s.t.} \quad (33e). \tag{38}$$

Since the PS sets the denoising factor θ_m for each subcarrier m independently, problem $\mathcal{P}1.2$ can be decomposed into M independent subproblems. Each m-th subproblem is given by

$$\min_{\theta_m} \Omega_m = \sum_{k=1}^{K} \frac{a_{k,m}}{|S_m|^2} \left(\frac{\sqrt{p_{k,m}}|h_{k,m}|}{\sqrt{\theta_m}} - 1 \right)^2 + \frac{\sigma^2}{\theta_m |S_m|^2}, \tag{39a}$$

$$\text{s.t.} \quad \theta_m \geq 0. \tag{39b}$$

By letting $\beta_m = 1/\sqrt{\theta_m}$, problem (39) can be transformed into the following form

$$\min_{\beta_m \geq 0} \sum_{k=1}^{K} a_{k,m} \left(\sqrt{p_{k,m}}|h_{k,m}|\beta_m - 1 \right)^2 + \sigma^2 \beta_m^2. \tag{40}$$

It can be seen that problem (40) is a convex quadratic optimization problem with respect to β_m. The unique optimal solution for the denosing factor θ_m can then obtained as

$$\theta_m^* = \left(\frac{\sigma^2 + \sum_{k=1}^{K} \left(a_{k,m} \sqrt{p_{k,m}}|h_{k,m}| \right)^2}{\sum_{k=1}^{K} a_{k,m} \sqrt{p_{k,m}}|h_{k,m}|} \right)^2, \ \forall m \in \mathcal{M}. \tag{41}$$

C. Transmission Power Optimization

Given the known subcarrier allocation $\mathbf{A}$ and the denoising factor vector Θ, problem $\mathcal{P}1$ can be simplified to the following

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI: 10.1109/TMC.2025.3582683, IEEE Transactions on Mobile Computing

power allocation optimization problem.

$$\mathcal{P}1.3: \quad \min_{\mathbf{P}} \sum_{m=1}^{M}\sum_{k=1}^{K} \frac{a_{k,m}}{|S_m|^2}\left(\frac{\sqrt{p_{k,m}}|h_{k,m}|}{\sqrt{\theta_m}}-1\right)^2 \quad (42)$$
$$\text{s.t.} \quad (33b) \sim (33c)$$

Given the independence of power allocation across devices, problem $\mathcal{P}1.3$ can be decomposed into K distinct subproblems, each corresponding to a single device k:

$$\min_{p_{k,m}} \sum_{m=1}^{M} \frac{a_{k,m}}{|S_m|^2}\left(\frac{\sqrt{p_{k,m}}|h_{k,m}|}{\sqrt{\theta_m}}-1\right)^2, \ \forall k \in \mathcal{K} \quad (43a)$$
$$\text{s.t.} \quad 0 \le p_{k,m} \le P_k^{\max}, \ \forall m \in \mathcal{M}, \quad (43b)$$
$$0 \le \sum_{m=1}^{M} a_{k,m} p_{k,m} \le P_k^{\text{tol}}. \quad (43c)$$

This formulation represents a convex quadratic programming problem with respect to $\sqrt{p_{k,m}}$, subject to linear power budget constraints given by (43b) and (43c). It can be efficiently addressed using convex optimization techniques, such as those implemented in CVX [38].

D. The Overall Algorithm

To solve problem $\mathcal{P}1$, we address subproblems $\mathcal{P}1.1$, $\mathcal{P}1.2$, and $\mathcal{P}1.3$ sequentially. The overall procedure is summarized in **Algorithm 2**, and the convergence of the algorithm is assured by the following **Lemma 2**.

Algorithm 2: The proposed joint optimization algorithm for solving problem $\mathcal{P}1$.

Input : Channel matrix $\mathbf{H} = [h_{k,m}] \in \mathbb{R}^{K\times M}$
Output: $\{\mathbf{A}^*, \Theta^*, \mathbf{P}^*\}$

Initialization: Subcarrier allocation matrix $\mathbf{A}^{(0)}$, denoising factor $\Theta^{(0)}$, transmission power $\mathbf{P}^{(0)}$, iteration rounds N and $i = 0$
repeat
  Given $\left\{\Theta^{(i)}, \mathbf{P}^{(i)}\right\}$, update $\mathbf{A}^{(i+1)}$ by $\mathbf{Algorithm1}$;
  Given $\left\{\mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}\right\}$, update $\Theta^{(i+1)}$ by eq. (41);
  Given $\left\{\mathbf{A}^{(i+1)}, \Theta_m^{(i+1)}\right\}$, update $\mathbf{P}^{(i+1)}$ by solving $\mathcal{P}1.3$;
  $i \leftarrow i + 1$;
until $i \ge N$;

Lemma 2. *The objective value of $\mathcal{P}1$ consistently decreases throughout the iteration of **Algorithm 2** and ultimately converges to a stable point.*

Proof. For a detailed proof, please refer to Appendix D. □

Next, we analyze the computational complexity of **Algorithm 2**. Given that the complexity of **Algorithm 1** is $\mathcal{O}(K^2)$, solving problem $\mathcal{P}1.1$ has a complexity of $\mathcal{O}(K^2M)$. The complexity of computing the denoising factor Θ is $\mathcal{O}(M)$. Problem $\mathcal{P}1.3$ is solved using CVX, which typically employs the interior point method for convex quadratic programming [39], resulting in a complexity of $\mathcal{O}(KM^{2.5})$. Therefore, the overall computational complexity of **Algorithm 2** is bounded by $\mathcal{O}(K^2M^{2.5})$. Since M (the number of data classes) and K (the number of devices) are relatively small, **Algorithm 2** can be efficiently executed in real-time on temporarily deployed edge servers with sufficient communication and computational resources.

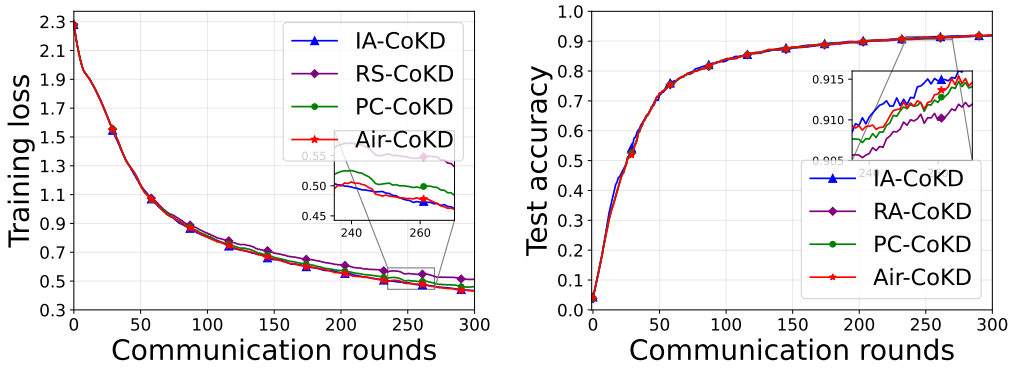
Finally, combining the framework of `Air-CoKD` introduced in Sect. IV-C, **Algorithm 2** is executed as follows within the `Air-CoKD` workflow: At the start of each training round t, the PS acquires the channel coefficient $h_{k,m}^t$ of device k on subcarrier m via channel measurement and estimation. It then determines the approximate optimal subcarrier allocation $\mathbf{A}$, power allocation $\mathbf{P}$, and denoising factor Θ by executing **Algorithm 2**. In practical deployments, however, channel estimation errors may degrade the accuracy of logits aggregation. Specifically, the estimation error introduces inaccuracy in the pre-processing coefficient $b_{k,m}^t$, which in turn leads to CSI-induced distortion in the aggregated result Ω_m^t [30], [40]. To mitigate this, the joint subcarrier assignment and power control strategy can be adapted to explicitly account for such CSI uncertainty. Nonetheless, for theoretical tractability and in line with common practice in related studies [21], [41], we adopt the idealized assumption of perfect CSI in our algorithmic analysis. After the resources allocation stage, devices use local data to update their models and generate local logits, which are modulated according to the power allocation $\mathbf{P}$ and transmitted over their assigned subcarriers $\mathbf{A}$. Upon receiving the aggregated logits, the PS applies post-processing using the denoising factor Θ, and broadcasts the processed logits back to the devices for the next training round. This process repeats until the maximum number of training rounds T is reached or the available energy at devices is exhausted.

VIII. Simulation Results

This section presents the simulation results for evaluating the `Air-CoKD` framework. The simulation parameters follow standard settings from the literature [20], [37], [42]. In the simulation, a PS and $K = 20$ devices are randomly placed within a circular area to model a resource-constrained network. The available bandwidth for CML is 1 MHz and is divided into M subcarriers, where M depends on the number of global data classes. Each subcarrier's bandwidth is denoted as B_{sub}. The channel coefficients $h_{k,m}^t$ follow Rayleigh fading, represented as $h_{k,m}^t \sim \mathcal{CN}(0,1)$, with flat fading assumed within each training round. The noise variance for AWGN is set to $\sigma^2 = 0.5$. We set the average power budget of any device k to be $P_k^{\text{tol}} = 10$ and the maximum power constraint to be $P_k^{\max} = 5$. The learning rate is $\eta = 0.05$, and the KD regularization coefficient is $\gamma_k = 1, \forall k \in \mathcal{K}$. Additional simulation parameters include:

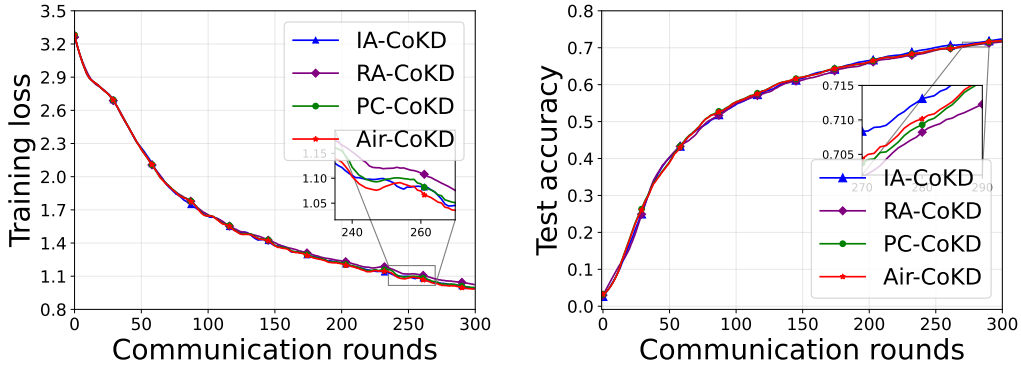
- *Dataset:* The MNIST, EMNIST, and CIFAR-10 datasets are used. MNIST has 70,000 28×28 images of handwritten digits, with 60,000 for training and 10,000 for testing. EMNIST refers to the EMNIST-letters subset, which

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI: 10.1109/TMC.2025.3582683, IEEE Transactions on Mobile Computing



(a) Training loss (b) Test accuracy

Fig. 2. Performance comparison on MNIST dataset.



(a) Training loss (b) Test accuracy

Fig. 3. Performance comparison on EMNIST dataset.

contains 28×28 grayscale images of handwritten English letters across 26 classes. CIFAR-10 includes 60,000 color images of size 32×32 spanning 10 object categories, with 50,000 training and 10,000 testing samples.

- *Data Distribution:* Non-IID distributions are simulated. The Dirichlet distribution, denoted as $Dir(\alpha)$, is used to partition the training dataset, where α is the concentration parameter. A smaller value of α results in higher data heterogeneity, whereas a larger α approximates the IID setting. We set $\alpha = 1.0$ for the Dirichlet distribution and distribute 50% of the training dataset to all devices.
- *Training Model:* The 7-layer CNN includes two 5×5 convolutional layers with 32 and 64 channels, followed by max pooling, two fully connected layers with 512 and 10 units, and a softmax output layer.

A. Performance Evaluation of `Air-CoKD`

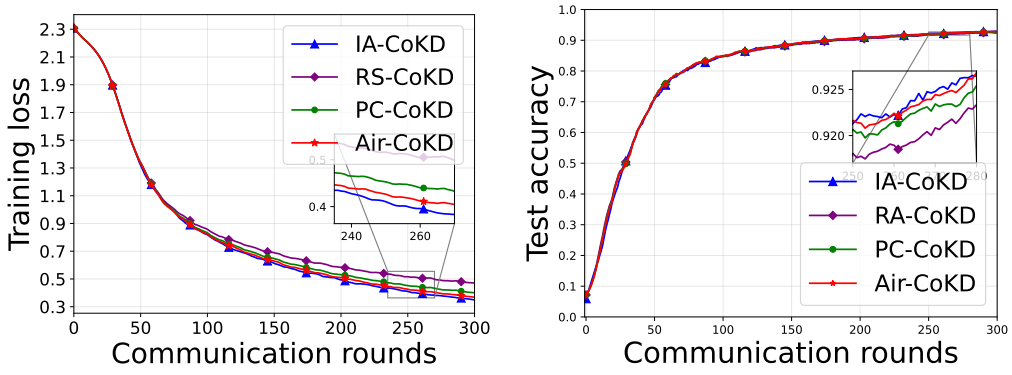
This section evaluates the effectiveness of **Algorithm 2**, the joint optimization algorithm for communication and learning parameters in the `Air-CoKD` framework. To evaluate the proposed method, we implemented the following baseline algorithms that optimize partial parameters or not: (1) `IA-FedKD` [43]: A `FedKD` method with ideal aggregation, avoiding channel fading and AWGN interference; (2) `PC-CoKD` [42]: A variant of `Air-CoKD` focusing on device transmission power control with random subcarrier assignment; (3) `RA-CoKD`: Another simplified variant of `Air-CoKD` that adopts random subcarrier and power allocation. We evaluated the model performance on datasets with distinct data classes. Specifically, MNIST comprises 10 handwritten digits classes, while EMNIST includes 26 English letter classes.

Figs. 2 and Figs. 3 show training loss and test accuracy for MNIST and EMNIST datasets, respectively. With 300 global communication rounds and local SGD for model updates, `IA-FedKD` performs best due to avoiding channel attenuation and AWGN, resulting in accurate global logits. It can be seen that the `Air-CoKD` method nearly matches `IA-FedKD` in performance, demonstrating high convergence and efficacy. Conversely, `SA-CoKD` and `PC-CoKD`, optimizing only partial parameters or not, show inferior performance. The `Air-CoKD` framework, utilizing OFDM for logits aggregation, effectively mitigates data class imbalance challenges by optimizing subcarrier allocation and communication strategies. This results in reduced aggregation errors and improved model performance compared to `SA-CoKD` and `PC-CoKD`, although it does not reach the ideal performance of `IA-FedKD`.

Fig. 4 and Fig. 5 evaluates the proposed `Air-CoKD` under the data distribution $\alpha = 10$, which indicates a distribution that approximates the IID setting. Compared to the results in Figs. 4 and 5, it is clear that model performances under data distribution $\alpha = 1.0$ decreases as data heterogeneity increases, especially on the EMNIST dataset. This can be attributed to the larger number of data classes in EMNIST dataset, where higher heterogeneity hinders device model from acquiring sufficient knowledge from local dataset.

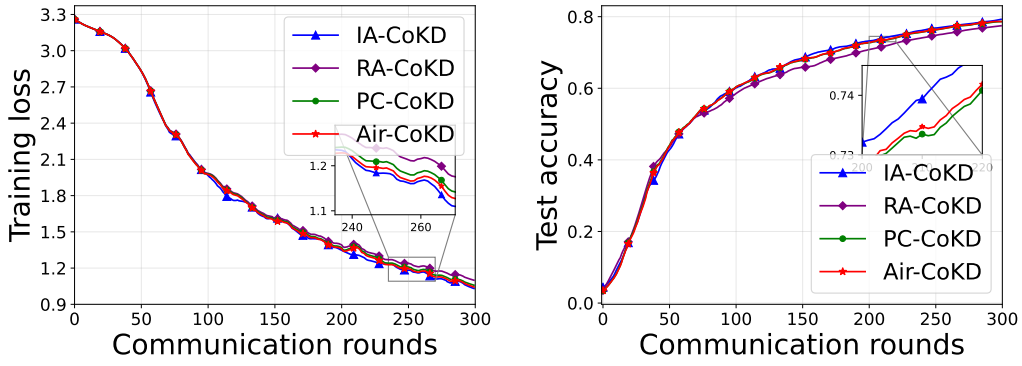
To evaluate model performance with varying numbers of participating devices, the training samples of the MNIST dataset are distributed across 40 devices, and performance

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI: 10.1109/TMC.2025.3582683, IEEE Transactions on Mobile Computing



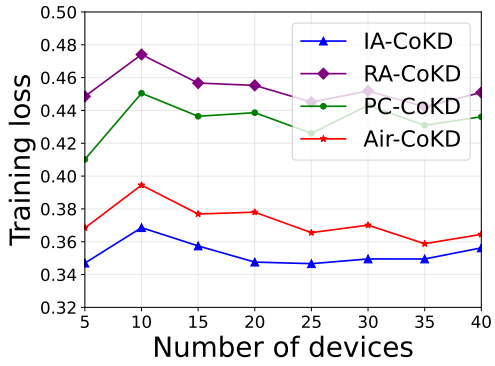
(a) Training loss (b) Test accuracy

Fig. 4. Performance comparison on MNIST dataset under $\alpha = 10$.

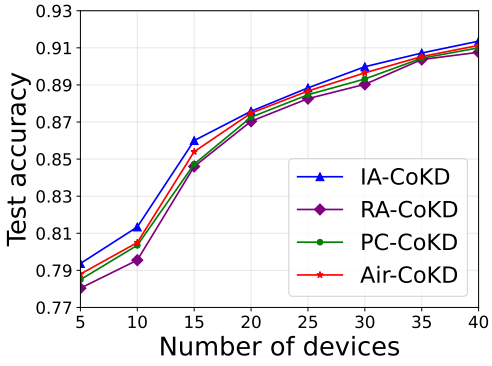


(a) Training loss (b) Test accuracy

Fig. 5. Performance comparison on EMNIST dataset under $\alpha = 10$.



(a) Trainning loss



(b) Test accuracy

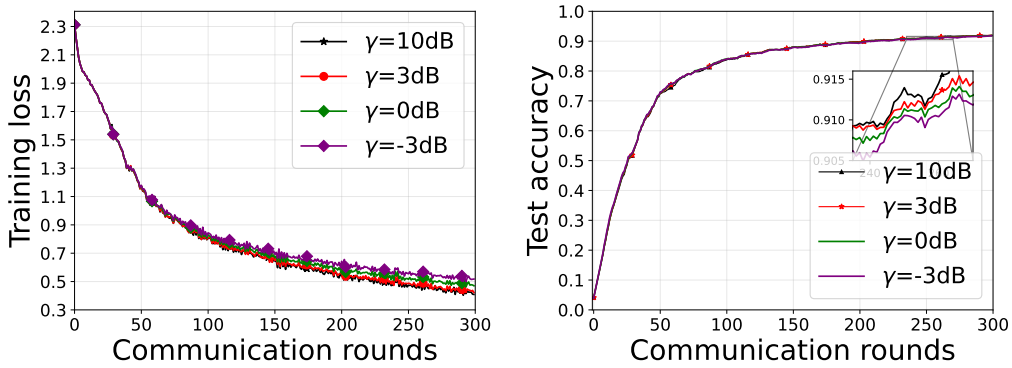
Fig. 6. Performance evaluation with varying numbers of devices.

is assessed as the number of participants increases. Initially, as more devices contribute, the utilization of additional data significantly enhances model performance. However, when the number of active devices becomes sufficiently large, performance stabilizes due to data redundancy, where additional devices provide overlapping data that no longer improves training effectiveness.

Then, we evaluate the convergence performance of `Air-CoKD` under different channel conditions. We approximate varying channel conditions by adjusting the channel-to-noise ratio (CNR), denoted as $\gamma = \frac{E\left[|h_k|^2\right]}{\sigma^2}, \quad \forall k \in \mathcal{K}$. The convergence of model performance, including training loss and test accuracy on the MNIST dataset, under different values of CNR (γ) is illustrated in Fig. 7. With varying values of γ, both training loss and test accuracy exhibit trends of local convergence, with diminishing returns as convergence progresses. Furthermore, as γ increases (i.e., as the CNR improves), the model demonstrates enhanced performance in terms of both training loss and test accuracy. More importantly, despite variations in CNR, the proposed `Air-CoKD` method exhibits strong robustness under different channel conditions, indicating its adaptability in communication environments with limited resources.

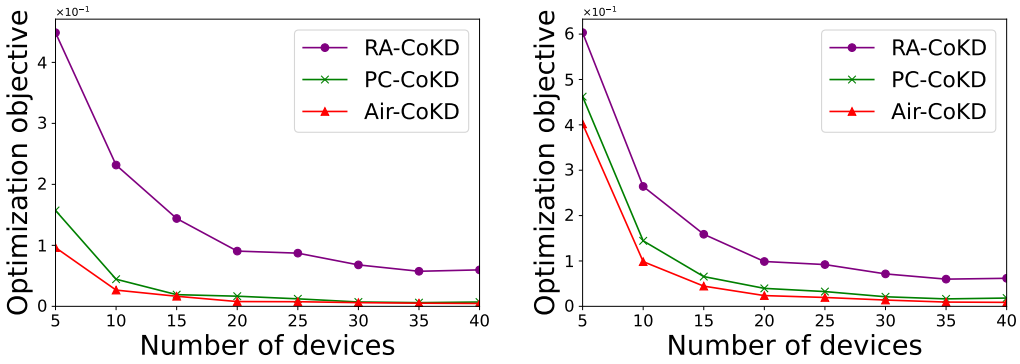
Fig. 8 compares the optimization objective, defined in eq. (35) of problem $\mathcal{P}1$, achieved by various algorithms under different noise conditions. This objective, representing the MSE of the aggregated logits (Ω_m) across all subcarriers, is derived from the model convergence analysis upper bound. As shown, `Air-CoKD` consistently achieves the lowest MSE,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI: 10.1109/TMC.2025.3582683, IEEE Transactions on Mobile Computing



(a) Trainning loss (b) Test accuracy

Fig. 7. Performance evaluation under different channel condition.



(a) $\sigma^2 = 0.1$ (b) $\sigma^2 = 0.5$

Fig. 8. Optimization objective vs. noise conditions for various algorithms.

effectively minimizing aggregation errors. The objective value increases with the noise level, indicating a positive correlation. As the number of participating devices increases, the objective value gradually decreases and eventually stabilizes, enhancing flexibility in parameter adjustments, particularly in subcarrier allocation.

Overall, the simulation results confirm that the proposed `Air-CoKD` framework excels in capturing the benefits of knowledge aggregation, enhancing model accuracy and convergence speed.

B. Advantages of `Air-CoKD` in Training Efficiency

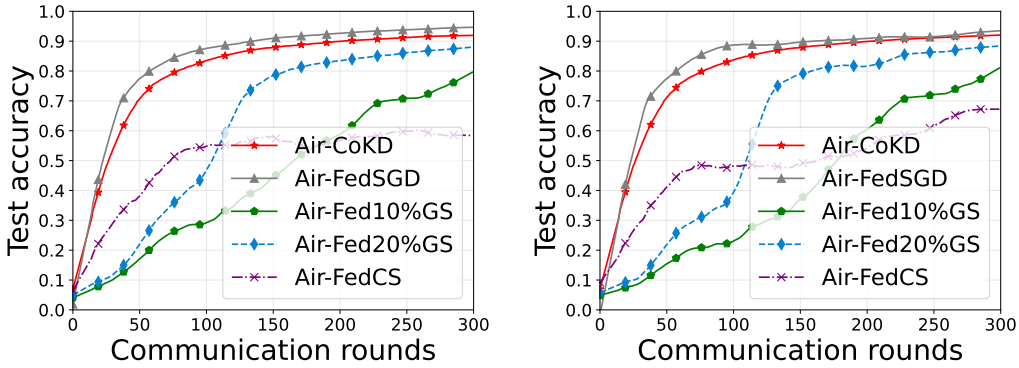
This section validates the learning time and resource utilization efficiency of the proposed `Air-CoKD`. To this end, we conduct experiments on the CIFAR-10 dataset and compare `Air-CoKD` with other AirComp-based model aggregation algorithms, including: (1) `Air-FedSGD`, which uses AirComp to transmit full model parameters; (2) `Air-FedGS`, a gradient sparsity-based method with random sparsity matrices, evaluated at 20% and 10% local parameter retention as `Air-Fed20%GS` and `Air-Fed10%GS`; (3) `Air-FedCS` [25], which combines gradient sparsity and one-bit quantization, transmitting 1,000 parameters. All methods optimize device transmission power and receiver denoising factors, while `Air-CoKD` additionally jointly optimizes subcarrier allocation and power control.

Figs. 9 and 10 show the convergence performance in terms of test accuracy on the MNIST and CIFAR-10 datasets, respectively, under different noise levels. While `Air-FedSGD` achieves the highest accuracy due to full gradient transmission, it is bandwidth-intensive. In contrast, `Air-FedGS` and `Air-FedCS` reduce communication latency but suffer from decreased accuracy and slower convergence, especially under high gradient sparsity. Our proposed method, `Air-CoKD`, effectively balances convergence speed and accuracy, demonstrating strong potential in resource-constrained environments. By comparing Figs. 9(a) and 9(b) (or Figs. 10(a) and 10(b)), it is evident that increasing the noise variance σ^2 leads to a general decline in convergence speed, test accuracy, and stability across all methods. Nevertheless, `Air-CoKD` consistently maintains test accuracy within an acceptable range, exhibiting stronger robustness to channel noise. In contrast, methods relying directly on AirComp-based gradient aggregation are more susceptible to gradient distortion under high noise, with performance degradation approaching 10 percentage points. This instability stems from gradient deviations introduced by channel noise, which impair model convergence and result in significant accuracy fluctuations. These findings highlight the noise robustness of our proposed `Air-CoKD` framework.

To evaluate model performance under a different data distribution, Fig. 11 presents a performance comparison of various CML methods with the Dirichlet parameter $\alpha = 10$. The results demonstrate that `Air-CoKD` consistently delivers competitive accuracy with significantly reduced communication overhead, underscoring its suitability for resource-constrained networks. Furthermore, these findings highlight the robustness of `Air-CoKD` under varying degrees of data heterogeneity.

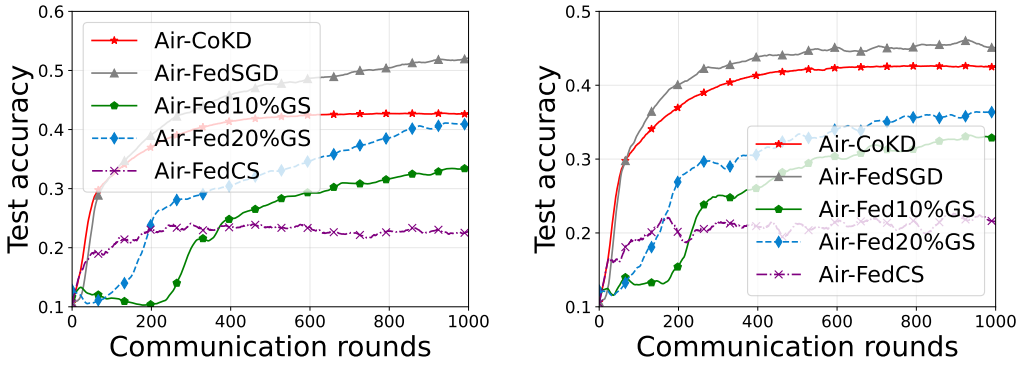
Analog modulation is employed during AirComp based

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI: 10.1109/TMC.2025.3582683, IEEE Transactions on Mobile Computing



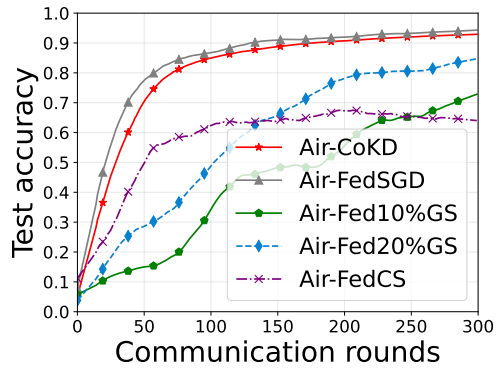
(a) $\sigma^2 = 0.1$ (b) $\sigma^2 = 0.5$

Fig. 9. Convergence performance of test accuracy under different noise conditions on MNIST dataset.

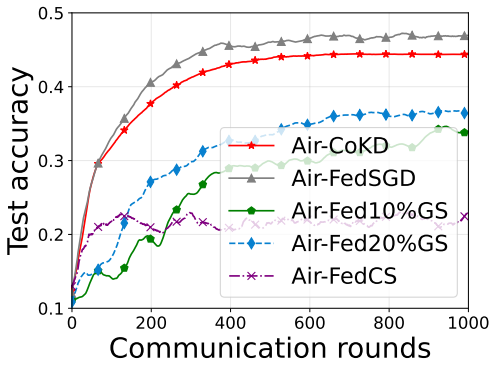


(a) $\sigma^2 = 0.1$ (b) $\sigma^2 = 0.5$

Fig. 10. Convergence performance of test accuracy under different noise conditions on CIFAR-10 dataset.



(a) MNIST dataset



(b) CIFAR-10 dataset

Fig. 11. Learning performance comparison versus different AirComp based CML methods under $\alpha = 10$.

TABLE I
TRAINING EFFICIENCY AND TEST ACCURACY OF DIFFERENT METHODS.

	MNIST(%)	CIFAR-10(%)	Parameters	Time (ms)
Air-FedSGD	93.21	45.65	582,026/583626	582.0/583.6
Air-Fed10%GS	79.23	32.50	58,203/58363	58.2/58.4
Air-Fed20%GS	87.80	36.48	116,406/116725	116.4/116.7
Air-FedCS	67.76	22.34	1,000	1.0
Air-CoKD	92.31	42.50	**100**	**0.1**

aggregation, where each parameter is amplitude-modulated to an analog symbol, with each subcarrier dedicated to a single parameter. The communication interval for AirComp aggregation is calculated as $\Delta t = \text{ceil}\left(\frac{p}{M}\right)\tau$, where $\text{ceil}(\cdot)$ is the ceiling function, M is the number of subcarriers, p is the size of transmitted information, and $\tau = \frac{1}{B_{\text{sub}}}$ is the OFDM symbol duration. For the MNIST and CIFAR-10 datasets, $M = 10$ and $B_{\text{sub}} = 0.1$ MHz. The time cost of CP is ignored for simplicity.

Table I highlights the uplink communication costs per round, where `Air-CoKD` transmits only local logits ($\phi_m \in \mathbb{R}^{10}$) for MNIST and CIFAR-10 datasets, achieving significant savings in both time and data transmission. It significantly reduces transmitted parameters compared to `Air-FedSGD`, `Air-FedGS`, and `Air-FedCS`, lowering transmission demands by several orders of magnitude.

Table II compares the average energy consumption of devices on the MNIST and CIFAR-10 dataset over 100 communication rounds. `Air-CoKD` outperforms `RA-CoKD` and `PC-CoKD`, achieving the best performance with the lowest energy consumption due to its optimized subcarrier allocation and power control strategies. This efficiency minimizes aggre-

TABLE II
AVERAGE ENERGY CONSUMPTION OF DEVICES.

	MNIST dataset	CIFAR-10 dataset
RA-CoKD	972	983
PC-CoKD	937	938
Air-CoKD	**878**	**883**

gation errors and reduces device-side power consumption.

Air-CoKD consistently balances communication efficiency and model performance across datasets and data distributions, making it particularly suitable for resource-constrained devices. These results underscore its applicability in energy-efficient and resource-limited networks.

IX. CONCLUSION

This paper investigates the Air-CoKD framework, which integrates KD with AirComp to enhance AI model training in resource-limited networks. An OFDM-based logits aggregation strategy is designed to address unbalanced device data classes, and closed-form expressions for convergence upper bounds in convex and non-convex scenarios are derived to illustrate the impact of logits aggregation errors on learning performance. A joint subcarrier allocation and power control method is proposed to minimize the convergence upper bound. Simulation results on MNIST, EMNIST, and CIFAR-10 datasets demonstrate that Air-CoKD achieves low communication delay while maintaining acceptable learning performance, outperforming baseline methods. These results highlight Air-CoKD as an effective solution for balancing training efficiency and performance in resource-limited networks.

Several unexplored aspects remain and are planned for future work: 1) Addressing model heterogeneity by exploring inter-device aggregation mechanisms to accommodate diverse neural network architectures caused by data heterogeneity; 2) Incorporating differential privacy (DP) mechanisms, such as privacy-aware device power control, to enhance privacy protection while maintaining performance.

REFERENCES

[1] R. H. L. Sim, Y. Zhang, M. C. Chan, and B. K. H. Low, "Collaborative machine learning with incentive-aware model rewards," in *Proc. 37th Int. Conf. Mach. Learn. (ICML)*, Jul. 2020, pp. 8927–8936.

[2] Z. Jiang, Y. Xu, H. Xu, Z. Wang, J. Liu, C. Qian, and C. Qiao, "Computation and communication efficient federated learning with adaptive model pruning," *IEEE Trans. Mob. Comput.*, vol. 23, no. 3, pp. 2003–2021, Mar. 2024.

[3] Y. Li, X. Qin, K. Han, N. Ma, X. Xu, and P. Zhang, "Accelerating wireless federated learning with adaptive scheduling over heterogeneous devices," *IEEE Internet Things J.*, vol. 11, no. 2, pp. 2286–2302, Jan. 2024.

[4] M. M. Amiri and D. Gündüz, "Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air," *IEEE Trans. Signal Process.*, vol. 68, pp. 2155–2169, Mar. 2020.

[5] Q. Wu, W. Wang, P. Fan, Q. Fan, H. Zhu, and K. B. Letaief, "Cooperative edge caching based on elastic federated and multi-agent deep reinforcement learning in next-generation networks," *IEEE Trans. Netw. Serv. Manag.*, vol. 21, no. 4, pp. 4179–4196, Aug. 2024.

[6] C. Zhang, W. Zhang, Q. Wu, P. Fan, Q. Fan, J. Wang, and K. B. Letaief, "Distributed deep reinforcement learning based gradient quantization for federated learning enabled vehicle edge computing," *IEEE Internet Things J.*, 2024, DOI: 10.1109/JIOT.2024.3447036.

[7] Z. Shao, Q. Wu, P. Fan, N. Cheng, W. Chen, J. Wang, and K. B. Letaief, "Semantic-aware spectrum sharing in internet of vehicles based on deep reinforcement learning," *IEEE Internet Things J.*, vol. 11, no. 23, pp. 38 521–38 536, Dec. 2024.

[8] G. Pang, "Artificial intelligence for natural disaster management," *IEEE Intell. Syst.*, vol. 37, no. 6, pp. 3–6, Nov. 2022.

[9] Z. Su, Y. Wang, Q. Xu, and N. Zhang, "LVBS: lightweight vehicular blockchain for secure data sharing in disaster rescue," *IEEE Trans. Dependable Secur. Comput.*, vol. 19, no. 1, pp. 19–32, Jan. 2022.

[10] M. Adam, A. Albaseer, U. Baroudi, and M. Abdallah, "Survey of multimodal federated learning: Exploring data integration, challenges, and future directions," *IEEE Open J. Commun. Soc.*, vol. 6, pp. 2510–2538, Mar. 2025.

[11] J. Du, T. Lin, C. Jiang, Q. Yang, F. Bader, and Z. Han, "Distributed foundation models for multi-modal learning in 6g wireless networks," *IEEE Wirel. Commun.*, vol. 31, no. 3, pp. 20–30, Jun. 2024.

[12] A. Mora, I. Tenison, P. Bellavista, and I. Rish, "Knowledge distillation for federated learning: a practical guide," *arXiv:2211.04742*, 2022.

[13] 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.

[14] 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.

[15] Z. Chen, W. Yi, Y. Liu, and A. Nallanathan, "Communication-efficient federated learning with heterogeneous devices," in *Proc. IEEE Int. Conf. Commun. (ICC), Rome, Italy*, May. 2023, pp. 3602–3607.

[16] T. Liu, J. Xia, Z. Ling, X. Fu, S. Yu, and M. Chen, "Efficient federated learning for aiot applications using knowledge distillation," *IEEE Internet Things J.*, vol. 10, no. 8, pp. 7229–7243, Apr. 2023.

[17] R. Mishra, H. P. Gupta, and T. Dutta, "A network resource aware federated learning approach using knowledge distillation," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS), Vancouver, BC, Canada*, May. 2021, pp. 1–2.

[18] Z. Zhu, J. Hong, and J. Zhou, "Data-free knowledge distillation for heterogeneous federated learning," in *Proc. 38th Int. Conf. Mach. Learn. (ICML)*, Jul. 2021, pp. 12 878–12 889.

[19] Y. Deng, J. Ren, C. Tang, F. Lyu, Y. Liu, and Y. Zhang, "A hierarchical knowledge transfer framework for heterogeneous federated learning," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM), NY, USA*, May. 2023, pp. 1–10.

[20] X. Cao, G. Zhu, J. Xu, Z. Wang, and S. Cui, "Optimized power control design for over-the-air federated edge learning," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 342–358, Jan. 2022.

[21] 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.

[22] W. Guo, R. Li, C. Huang, X. Qin, K. Shen, and W. Zhang, "Joint device selection and power control for wireless federated learning," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 8, pp. 2395–2410, Aug. 2022.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] O. Shahid, S. Pouriyeh, R. M. Parizi, Q. Z. Sheng, G. Srivastava, and L. Zhao, "Communication efficiency in federated learning: Achievements and challenges," *arXiv:2107.10996*, 2021.

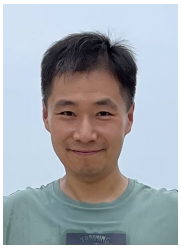
[28] M. Tan and Q. V. Le, "Efficientnetv2: Smaller models and faster training," in *Proc. 38th Int. Conf. Mach. Learn. (ICML)*, Jul. 2021, pp. 10 096–10 106.

[29] R. Anil, G. Pereyra, A. Passos, R. Ormándi, G. E. Dahl, and G. E. Hinton, "Large scale distributed neural network training through online distillation," *arxiv/1804.03235*, 2018.

[30] Y. Chen, H. Xing, J. Xu, L. Xu, and S. Cui, "Over-the-air computation in OFDM systems with imperfect channel state information," *IEEE Trans. Commun.*, vol. 72, no. 5, pp. 2929–2944, May. 2024.

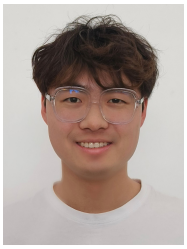
[31] P. Yang, D. Wen, Q. Zeng, Y. Zhou, T. Wang, H. Cai, and Y. Shi, "Over-the-air computation empowered vertically split inference," *IEEE Trans. Wirel. Commun.*, vol. 23, no. 12, pp. 19 634–19 648, Dec. 2024.
[32] Y. Yang, Y. Zhou, Y. Wu, and Y. Shi, "Differentially private federated learning via reconfigurable intelligent surface," *IEEE Internet Things J.*, vol. 9, no. 20, pp. 19 728–19 743, Oct. 2022.
[33] D. Yao, W. Pan, Y. Dai, Y. Wan, X. Ding, H. Jin, Z. Xu, and L. Sun, "Local-global knowledge distillation in heterogeneous federated learning with non-iid data," *arXiv:2107.00051*, 2021.
[34] 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.
[35] A. Jacot, C. Hongler, and F. Gabriel, "Neural tangent kernel: Convergence and generalization in neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS), Montréal, Canada*, Dec. 2018, pp. 8580–8589.
[36] N. Yan, K. Wang, C. Pan, K. K. Chai, F. Shu, and J. Wang, "Over-the-air federated averaging with limited power and privacy budgets," *IEEE Trans. Commun.*, vol. 72, no. 4, pp. 1998–2013, Apr. 2024.
[37] D. Wang, N. Zhang, M. Tao, and X. Chen, "Knowledge selection and local updating optimization for federated knowledge distillation with heterogeneous models," *IEEE J. Sel. Top. Signal Process.*, vol. 17, no. 1, pp. 82–97, Jan. 2023.
[38] S. Diamond and S. P. Boyd, "CVXPY: A python-embedded modeling language for convex optimization," *J. Mach. Learn. Res.*, vol. 17, pp. 83:1–83:5, 2016.
[39] S. Bubeck, "Convex optimization: Algorithms and complexity," *Found. Trends Mach. Learn.*, vol. 8, no. 3-4, pp. 231–357, 2015.
[40] B. Jiang, J. Du, C. Jiang, Z. Han, A. Alhammadi, and M. Debbah, "Over-the-air federated learning in digital twins empowered UAV swarms," *IEEE Trans. Wirel. Commun.*, vol. 23, no. 11, pp. 17 619–17 634, Nov. 2024.
[41] N. Yan, K. Wang, C. Pan, K. K. Chai, F. Shu, and J. Wang, "Over-the-air federated averaging with limited power and privacy budgets," *IEEE Trans. Commun.*, vol. 72, no. 4, pp. 1998–2013, Apr. 2024.
[42] 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.
[43] H. Seo, J. Park, S. Oh, M. Bennis, and S.-L. Kim, "Federated knowledge distillation," *arXiv:2011.02367*, 2020.

Yue Zhang received the B.S. degree from the Changzhou Institute of Technology, Changzhou, China, in 2018 and the M.S. degrees from the 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.

Kun Yang received his PhD from the Department of Electronic & Electrical Engineering of University College London (UCL), UK. He is currently a Chair Professor in the School of Computer Science & Electronic Engineering, University of Essex, leading the Network Convergence Laboratory (NCL), UK. He is also an affiliated professor at UESTC, China. Before joining in the University of Essex at 2003, he worked at UCL on several European Union (EU) research projects for several years. His main research interests include wireless networks and communications, IoT networking, data and energy integrated networks and mobile computing. He manages research projects funded by various sources such as UK EPSRC, EU FP7/H2020 and industries. He has published 400+ papers and filed 30 patents. He serves on the editorial boards of both IEEE (e.g., IEEE TNSE, IEEE ComMag, IEEE WCL) and non-IEEE journals (e.g., Deputy EiC of IET Smart Cities). He was an IEEE ComSoc Distinguished Lecturer (2020-2021). He is a Member of Academia Europaea (MAE), a Fellow of IEEE, a Fellow of IET and a Distinguished Member of ACM.



Yao Wen received the B.S. degree from the College of Information Science and Technology, Nanjing Forestry University, Nanjing, China, in 2021, and the M.S. degree from the School of Computer Science and Technology, China University of Mining and Technology, Xuzhou, China, in 2024. He is pursuing the Ph.D. degree at Nanjing University. His research interests include mobile edge computing and wireless communications.

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.