

AirComp-Assisted Asynchronous Federated Learning for UAV Swarms: A Self-Adaptive Aggregation Scheme to Tackle Model Staleness

Yansong Huang^{1,2}, Xuan Li^{1*}, *Senior Member, IEEE*, Lu Zhang¹, Mugen Peng¹, *Fellow, IEEE*

¹ Beijing University of Posts and Telecommunications, China

² University College London, United Kingdom

Abstract—Federated learning (FL) is a promising paradigm for collaborative intelligence in the low-altitude economy, enabling unmanned aerial vehicle (UAV) swarms to perform deep learning tasks (e.g., logistics, emergency rescue) while preserving data sovereignty. However, limited communication channels and heterogeneous computation capabilities among UAVs cause significant FL aggregation delays. To reduce convergence time, we propose an asynchronous FL (AFL) framework for UAV swarms, which integrates over-the-air computation (AirComp) to improve communication efficiency via simultaneous transmission. To address signal distortion in AirComp, we formulate an objective function and solve it using an aggregation scheduling algorithm, which transforms the nonconvex problem into two convex subproblems tackled via alternating optimization, to derive optimal aggregation strategies and beamforming vectors. Moreover, to mitigate model staleness in AFL, which causes gradient divergence and slow convergence, we propose a self-adaptive aggregation scheme with staleness awareness, enabling UAVs to adjust local models autonomously without requiring information from other UAVs. Simulation results show that our scheme reduces staleness impact and leverages stale parameters, helping AFL outperform synchronous FL in convergence speed and accuracy. Overall, our study presents an effective AFL framework, a fast aggregation scheduling algorithm, and a self-adaptive aggregation scheme for UAV swarms, accelerating global model convergence while reducing energy expenditure.

Index Terms—Asynchronous federated learning, over-the-air computation, self-adaptive aggregation, UAV.

I. INTRODUCTION

A. Background

As an emerging economic sector, low-altitude economy, where the unmanned aerial vehicle (UAV) technology plays a key role [1]–[3], has led to its widespread application in various domains such as urban air mobility [4], emergency rescue [5], and drone logistics [6]. However, concerns about data security and the phenomenon of data isolation are impeding the further development of the low-altitude economy [7]. In this context, federated learning (FL) has emerged as an effective decentralized machine learning framework [8]–[10], where edge nodes train deep learning models locally using

This work was supported by the National Natural Science Foundation of China under Grant 62401076. The authors are with the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: xuan.li@bupt.edu.cn).

sensitive data and only share model updates with a central server for model aggregation, contributing to a more powerful global model that integrates knowledge from all local models while preserving data sovereignty. Iteratively, the edge nodes receive the global model as the baseline for subsequent deep learning training and as a tool for downstream tasks [11], [12].

However, traditional FL frameworks, namely synchronous federated learning (SFL), face several challenges in UAV swarm networks, particularly communication efficiency and computation capability [13], [14]. The limited bandwidth and energy resources of UAVs lead to the delay of model transmission and differences in computational capabilities among UAVs result in longer training time, which dramatically increases aggregation time under the constraint of aggregating all local models in each global epoch [15]. Asynchronous federated learning (AFL) appears to be a potential solution to this problem because it relaxes the constraint and allows the parameter server (PS) to only aggregate a subset of local models instead of all of them [16], [17]. As shown in Fig. 1, transmission delays and varying training durations lead to longer idle time in SFL than in AFL, resulting in longer aggregation time [18].

Nevertheless, while AFL effectively shortens iteration cycles, it introduces a critical challenge, namely model staleness [19]. Specifically, because some nodes fail to participate in global aggregation, their local models develop a version gap relative to the latest global model and thus these local models become stale [20]. These nodes may continue training on these stale models, which may gradually lead to significantly worse performance compared to those with fresh local models. When these stale models are aggregated, stale model parameters can distort the gradient update direction of the global model in subsequent aggregations, leading to convergence oscillations or even divergence [21], as shown in Fig. 2.

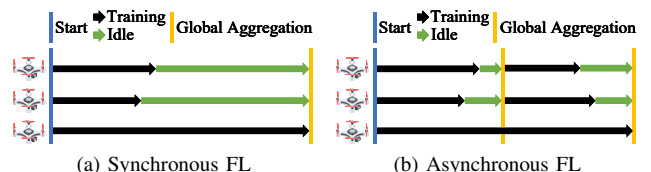


Fig. 1. AFL relaxes the aggregation constraint to save idle time and shorten the aggregation duration compared to SFL.

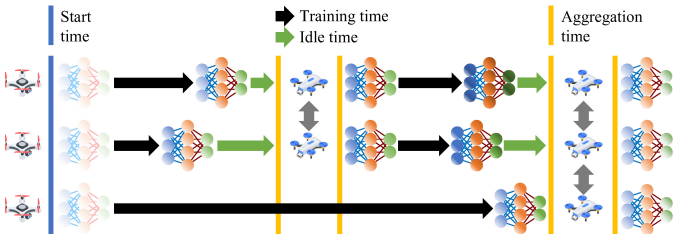


Fig. 2. Illustration of model staleness in AFL. Models shown in lighter colours indicate less knowledge, whereas darker colours indicate better performance.

B. Motivation

In scenarios such as emergency rescue, UAVs are allocated to collect data for situation analysis [22]. When deep learning is introduced into such tasks to reduce human workload, it is expected to support faster decision-making due to the urgency, even at some cost of accuracy [23]. However, when a deep learning model is applied to a new scenario, it usually needs to be fine-tuned to adapt itself to this scenario by training on datasets collected from ongoing incidents. In this case, the convergence time of the global model in FL is expected to be shorter, while maintaining model accuracy.

To address slow convergence, AFL has been proposed to relax the aggregation condition [16]. In the AFL scheme, the PS aggregates parameters from accessible UAVs at fixed intervals, rather than waiting for all UAVs to transmit their parameters [17]. By eliminating inefficient waiting time, iteration duration is significantly reduced. However, this also reduces the amount of training data available for updating the model. Research has shown that the decrease in training data per iteration results in lower accuracy of the final model [24]. Additionally, the model staleness issue in AFL, caused by delayed global model updates in UAVs, may result in slow improvement in model performance and even weight divergence [25]. To address the drawbacks of decreased training data and model staleness, an effective AFL framework should be designed to aggregate as many local models as possible within a given time, while also ensuring global model performance by mitigating staleness.

In order to further accelerate model aggregation, an advanced over-the-air computation (AirComp) transmission scheme may be integrated into the FL framework to utilize limited UAV channel bandwidth more efficiently and enable simultaneous communication and computation [26], [27]. It leverages signal superposition of weighted model parameters in the wireless multiple-access channel, thereby reducing energy consumption by integrating the reception of multiple local models [28]. However, despite the fact that AirComp utilizes signal superposition to avoid inter-channel interference, the signal is still affected by other factors, resulting in distortion that is difficult for the receiver to detect and correct, since it only receives the superposed signals from multiple transmitters with mixed distortions [29]–[31]. Hence, the beamforming vector of the receiver UAV should be carefully designed to eliminate channel interference from each transmitter UAV.

Sometimes, UAVs with poor channel conditions must be excluded from global aggregation in a given epoch to ensure AirComp transmission quality. However, this contradicts the

goal of AFL, which aims to include as many UAVs as possible in each epoch to maximize training data utilization. To balance the requirements of AFL and AirComp, an algorithm should be designed to select UAVs whose channel conditions meet AirComp transmission requirements and maximize the volume of training data used in each epoch. Given the large scale of UAV swarm networks, the algorithm should also have low computational complexity.

C. Related Work

Existing research on FL combined with AirComp primarily focuses on terrestrial networks. These studies have demonstrated the effectiveness of these technologies in enhancing communication efficiency and reducing latency in edge learning scenarios [29]–[31]. However, their direct application to UAV networks is not straightforward. UAVs face unique challenges, including limited computational capabilities and unstable communication links that extend the time required to aggregate local models and consequently slow the convergence of global models.

Some studies have started to adapt FL techniques to UAV scenarios, acknowledging limitations such as constrained computational power on UAVs [8]–[10]. These studies have employed various methods to reduce the convergence time of global models, but their convergence efficiency is still limited by the SFL mechanisms. For instance, He et al. [9] utilized Stackelberg equilibrium to optimize the allocation of limited resources and improve the effectiveness of the FL models.

Previous efforts to integrate FL with UAVs have largely focused on tackling the challenges caused by unreliable UAV-to-UAV links and accelerating the convergence of FL models. However, these studies often do not adequately consider the constraints of limited computational resources available on UAVs or the need to optimize power consumption for on-edge model training, thereby limiting their applicability in real-world UAV environments. Moreover, these works have not yet explored the unique challenges and potential benefits of applying AirComp to AFL within UAV networks, which might impact performance and energy efficiency.

In terms of model staleness, delayed or outdated local updates may degrade convergence stability and final model performance. Existing approaches that aim to relieve model staleness typically adjust aggregation weights, learning rates, or momentum terms at the server based on delay or version gaps [21], [32]. These methods rely on the server having access to individual local updates and explicit staleness information, enabling fine-grained control over each client's contribution. While effective in conventional AFL settings, such server-side strategies are difficult to apply in AirComp-assisted FL, where the server only receives superposed signals rather than per-client model updates.

Another line of work alleviates staleness by selectively scheduling clients with fresher models or better communication conditions. By excluding highly stale clients from aggregation, these methods reduce harmful updates at the cost of sacrificing training data diversity [33], [34]. However, in large-scale UAV swarms with dynamic topology and limited

communication resources, aggressive client exclusion may significantly reduce the effective data volume per global update, which contradicts the objective of AFL to maximize learning efficiency within a fixed time budget.

In contrast to the above methods, our work addresses model staleness from a fundamentally different perspective tailored to AirComp-assisted FL. It enables the global model to exploit informative components from highly stale local models, which are often trained on larger datasets or for longer durations, while suppressing harmful divergence caused by outdated layers. As a result, our approach not only mitigates staleness-induced instability but also leverages stale updates as a source of useful knowledge. This differs from existing staleness-aware AFL methods that rely solely on server-side weighting or update suppression.

D. Contributions and Organization

We design an AirComp-assisted AFL framework for UAV swarm networks, while jointly optimizing the linkage scheme and beamforming vectors, and propose a self-adaptive aggregation scheme to mitigate the staleness problem in AFL. To be more specific,

- we propose an efficient AFL framework for UAV swarm networks, where all UAVs work as edge nodes, collecting data and training local models, and the swarm heads act as central PSs, aggregating global models and transmitting them to users for downstream use.
- We apply an energy-efficient AirComp-assisted transmission scheme to the AFL framework by superposing the signals of local models from multiple UAVs simultaneously, fully utilizing communication time and bandwidth, and formulate a data volume maximization problem under signal-distortion and power constraints.
- We propose a suboptimal solution by decomposing the original problem into a UAV selection subproblem and a beamforming design subproblem, which are solved by a branch-and-bound algorithm and an alternating optimization algorithm, respectively.
- We propose a self-adaptive aggregation scheme to address the staleness problem in AFL, enabling UAVs to upload only those layers in local models that have high cosine similarity with the latest global model, thereby accelerating global model convergence.
- We evaluate the proposed AirComp-assisted AFL framework by implementing a practical deep learning task and compare our design in terms of model accuracy, loss, convergence time, energy efficiency, and transmission data volume against conventional SFL systems with and without AirComp transmission.

The rest of our paper is organized as follows: first, Section II presents the AirComp-assisted AFL in UAV swarms, focusing on the model aggregation workflow and the optimization problem. Next, Section III introduces our optimal solution for linkage schemes using the branch-and-bound algorithm and our suboptimal solution for beamforming vectors using the alternating optimization algorithm. Then, Section IV details our self-adaptive aggregation scheme for addressing model

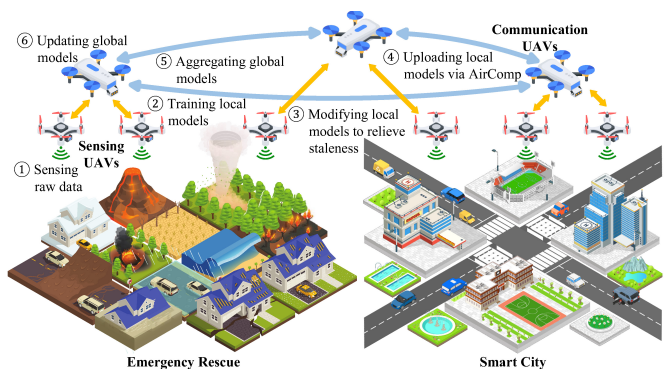


Fig. 3. Illustration of AFL in UAV swarms.

staleness in AFL. Subsequently, Section V presents the simulation results for various performance indicators. Finally, Section VI analyses the results and concludes our work.

II. SYSTEM MODEL

In this section, we introduce an AFL framework in UAV swarms, emphasizing the workflow of model aggregation. Additionally, we illustrate how to apply AirComp to AFL during the process of model aggregation to improve communication energy efficiency. The communication-quality constraint introduced by AirComp is modelled and described in detail.

A. AFL in UAV Swarms

Consider a UAV swarm consisting of S sensing UAVs $\mathcal{S} = \{s_1, s_2, \dots, s_S\}$ and C communication UAVs $\mathcal{C} = \{c_1, c_2, \dots, c_C\}$. A sensing UAV is equipped with sensors for data collection and one antenna for model transmission while a communication UAV is equipped with A antennas for model transmission. The trajectories of the UAVs are determined by specific path planning algorithms, which are beyond the scope of this paper.

As illustrated in Fig. 3, the communication UAVs in the swarm have stronger communication capabilities and are therefore connected with each other at all times. Meanwhile, the sensing UAVs are restricted to communicating only with the communication UAVs to reduce network complexity and interference. In this way, the sensing UAVs communicate with communication UAVs in a star topology while communication UAVs maintain interconnections through a backbone network for aggregating local models into global models and then synchronizing them. Such topology ensures the stability of the aggregation of local models under dynamic UAV mobility. Specifically, in the ongoing global epoch e , the sensing UAVs $s_i \in \mathcal{S}$ first collect data $\mathcal{D}^{[e]} = \{d_1^{[e]}, d_2^{[e]}, \dots, d_S^{[e]}\}$ and train local models $m_i^{[e]} \in \mathcal{M}^{[e]} = \{m_1^{[e]}, m_2^{[e]}, \dots, m_S^{[e]}\}$ based on their latest updated global model $m^{[e']}$ as

$$m_i^{[e]} = m^{[e']} - \eta \nabla \mathcal{J}_i \left(m^{[e']}, \left\{ d_i^{[e']}, \dots, d_i^{[e]} \right\} \right). \quad (1)$$

Here, e'_i refers to the prior global epoch when the sensing UAV s_i obtains the global model $m^{[e']}$ as the baseline model,

and typically $e'_i < e$. η is the learning rate, $\nabla \mathcal{J}_i(\cdot)$ denotes the gradient of the local loss function $\mathcal{J}_i(\cdot)$ of the UAV s_i . From global epoch e'_i to e , the sensing UAV s_i collects data $\{d_i^{[e'_i]}, \dots, d_i^{[e]}\}$ and trains its local model $m_i^{[e]}$ based on the global model $m^{[e'_i]}$.

Then, the communication UAVs \mathcal{C} determine the linkage scheme $\mathbf{L}^{[e]} \in \mathbb{R}^{C \times S}$ for this epoch, where every entry $l_{i,j}^{[e]} \in \mathbf{L}^{[e]}$ is defined as

$$l_{i,j}^{[e]} = \begin{cases} 1, & \text{if } s_i \text{ transmits to } c_j \text{ at epoch } e \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

subject to the constraint

$$\sum_{j=1}^C l_{i,j}^{[e]} \leq 1, \forall i = 1, 2, \dots, S. \quad (3)$$

The communication UAVs then send the latest global model $m^{[e-1]}$ to the sensing UAVs included in the UAV set $\mathcal{S}^{[e]} = \{s_i \in \mathcal{S} | \exists c_j \in \mathcal{C} \text{ where } l_{i,j} = 1\}$ which changes in each epoch according to the linkage scheme $\mathbf{L}^{[e]}$. The selected sensing UAVs $\mathcal{S}^{[e]}$ modify their local models $\mathcal{M}_{\text{upload}}^{[e]} = \{m_i^{[e]} \in \mathcal{M}^{[e]} | s_i \in \mathcal{S}^{[e]}\}$ using the global model $m^{[e-1]}$ to $\hat{\mathcal{M}}_{\text{upload}}^{[e]} = \{\hat{m}_i^{[e]} = f(m_i^{[e]}, m^{[e-1]}) | m_i^{[e]} \in \mathcal{M}_{\text{upload}}^{[e]}\}$ through the operation $f(\cdot)$. This process is part of the self-adaptive aggregation scheme, which will be detailed in Section IV. Then, the selected sensing UAVs $s_i \in \mathcal{S}^{[e]}$ upload local models $\mathcal{M}'_{\text{upload}}^{[e]}$ to communication UAVs for global aggregation

$$m^{[e]} = \left(1 - \frac{1}{\psi} \sum_{s_i \in \mathcal{S}^{[e]}} \rho_i\right) \cdot m^{[e-1]} + \frac{1}{\psi} \sum_{s_i \in \mathcal{S}^{[e]}} \rho_i \cdot \hat{m}_i^{[e]}. \quad (4)$$

Here, the preprocessing scalar ρ_i is defined as

$$\rho_i = \sum_{k=e'_i}^e \left| d_i^{[k]} \right| \cdot \frac{e'_i}{e} \quad (5)$$

whose first part $\sum_{k=e'_i}^e \left| d_i^{[k]} \right|$ indicates the volume of training data accumulated by sensing UAV s_i from epoch e'_i to current epoch e and the second part e'_i/e indicates the effect of model staleness. The postprocessing scalar ψ is defined as

$$\psi = \sum_{s_i \in \mathcal{S}^{[e]}} \sum_{k=e'_i}^e \left| d_i^{[k]} \right| \quad (6)$$

to normalize the model after applying the preprocessing scalar while excluding the staleness factor e'_i/e , whose effect is compensated by the previous global model $m^{[e-1]}$ following (4).

After receiving superposed local models from sensing UAVs through AirComp, communication UAVs \mathcal{C} select one of them to receive these superposed local models from the others and aggregate them into the new global model $m^{[e]}$. This process is still accomplished by AirComp, but the communication UAVs are less concerned about transmission distortion because of their stronger communication capability. Finally, the global model $m^{[e]}$ is sent to all communication UAVs \mathcal{C} and then forwarded to the reachable sensing UAVs $\mathcal{S}^{[e]}$ to continue the local training in the new epoch $e+1$.

Algorithm 1 AFL Workflow

Initialize global model $m^{[0]}$
 Communication UAVs \mathcal{C} broadcast $m^{[0]}$ to sensing UAVs \mathcal{S}
for epoch $e = 1, \dots, E$ **do**
 \mathcal{S} collect data $\mathcal{D}^{[e]}$ and train local models $\mathcal{M}^{[e]}$ as (1)
 \mathcal{C} determine linkage scheme $\mathbf{L}^{[e]}$ through Algorithm 2
 \mathcal{C} send global model $m^{[e-1]}$ to selected UAVs $\mathcal{S}^{[e]}$
 $\mathcal{S}^{[e]}$ modify local models $\mathcal{M}_{\text{upload}}^{[e]}$ into $\hat{\mathcal{M}}_{\text{upload}}^{[e]}$ based on Algorithm 4 and send them to \mathcal{C}
 \mathcal{C} aggregate local models $\hat{\mathcal{M}}_{\text{upload}}^{[e]}$ into a new global model $m^{[e]}$ as (4) and send to $\mathcal{S}^{[e]}$
end for

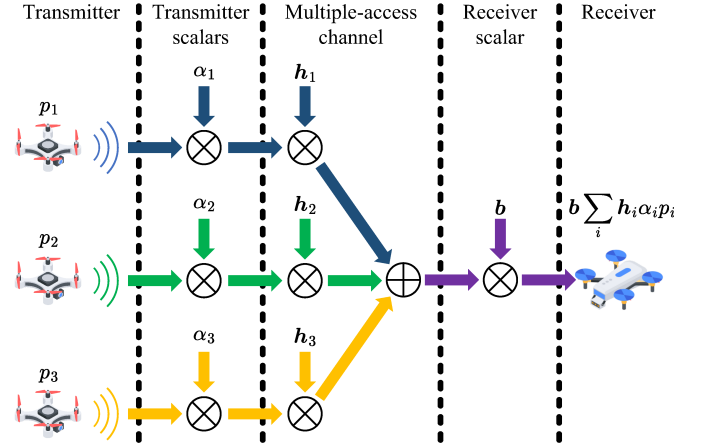


Fig. 4. Demonstration of AirComp.

To summarize, our proposed AFL framework includes four steps, data collection and training, modification, uploading and aggregation, and distribution. The details of the proposed AFL framework are illustrated in Algorithm 1.

B. AirComp for Model Aggregation

In the conventional FL framework, edge nodes transmit their local models to PSs separately, which limits communication efficiency under bandwidth constraints or results in higher energy consumption because of longer transmission time [35]. To fully utilize communication resources, AirComp is introduced to integrate computation and communication by exploiting the signal superposition property of the multiple-access channel and computing a linear function of distributed models from various UAVs, as illustrated in Fig. 4.

To upload the model $\hat{m}_i^{[e]}$ to the communication UAVs \mathcal{C} , UAVs $\mathcal{S}^{[e]}$ transmit the signal vector $\mathbf{p}_i := \hat{m}_i^{[e]} \in \mathbb{C}^T$, which is normalized to unit variance as $\mathbb{E}(\mathbf{p}_i \mathbf{p}_i^H) = \mathbf{I}$. Here, the operation $(\cdot)^H$ denotes the conjugate transpose of the original matrix. At time slot $t \in \{1, 2, \dots, T\}$, UAV $s_i \in \mathcal{S}^{[e]}$ sends the signal $p_i^{(t)}$ to UAV $c_j \in \mathcal{C}$ via AirComp, and the ideal received signal $p_j^{(t)}$ is denoted as

$$p_j^{(t)} = \sum_{s_i \in \mathcal{S}^{[e]}} \rho_i p_i^{(t)}. \quad (7)$$

For simplicity, the time slot index t is omitted in subsequent expressions. The effect of the multiple-access channel on the signal is given by

$$\mathbf{p}_j = \sum_{s_i \in \mathcal{S}^{[e]}} \mathbf{h}_{i,j} \alpha_i p_i + \mathbf{n}, \quad (8)$$

where \mathbf{p}_j is the signal affected by the multiple-access channel but not received by UAV c_j yet, $\mathbf{h}_{i,j} \in \mathbb{C}^A$ is the channel vector between UAV $s_i \in \mathcal{S}^{[e]}$ and $c_j \in \mathcal{C}$, $\alpha_i \in \mathbb{R}^+$ is the transmitter scalar, and $\mathbf{n} \sim \mathcal{CN}(\mathbf{0}, \sigma^2 \mathbf{I})$ is the noise vector where $\sigma > 0$ denotes the standard deviation of the noise. After reception through the beamforming vector $\mathbf{b}_j \in \mathbb{C}^A$ at UAV c_j , the expected signal \hat{p}_j is given by

$$\hat{p}_j = \frac{1}{\sqrt{\lambda_j}} \mathbf{b}_j^H \mathbf{p}_j = \frac{1}{\sqrt{\lambda_j}} \mathbf{b}_j^H \sum_{s_i \in \mathcal{S}^{[e]}} \mathbf{h}_{i,j} \alpha_i p_i + \frac{\mathbf{b}_j^H \mathbf{n}}{\sqrt{\lambda_j}} \quad (9)$$

where λ_j is a normalization factor. The power constraint at sensing UAVs $s_i \in \mathcal{S}$ is

$$|\alpha_i|^2 \leq P_S. \quad (10)$$

The constraint on the noise amplification factor at communication UAVs $c_j \in \mathcal{C}$ is

$$\|\mathbf{b}_j\|^2 \leq P_N. \quad (11)$$

The performance of AirComp is characterized by the distortion between the ideal signal p_j in (7) and the estimated signal \hat{p}_j in (9), which is measured by the mean-squared-error (MSE) defined as

$$\text{MSE}(\hat{p}_j, p_j) = \mathbb{E}(|\hat{p}_j - p_j|^2) \quad (12a)$$

$$= \sum_{s_i \in \mathcal{S}^{[e]}} \left| \frac{\mathbf{b}_j^H \mathbf{h}_{i,j} \alpha_i}{\sqrt{\lambda_j}} - \rho_i \right|^2 + \sigma^2 \frac{\|\mathbf{b}_j\|^2}{\lambda_j}. \quad (12b)$$

Given an arbitrarily receive beamforming vector \mathbf{b}_j , the optimal transmitter scalar that minimizes the MSE is obtained by the following zero-forcing condition [36]:

$$\frac{\mathbf{b}_j^H \mathbf{h}_{i,j} \alpha_i}{\sqrt{\lambda_j}} - \rho_i = 0. \quad (13)$$

Thus, the optimal transmitter scalar is

$$\alpha_i = \sqrt{\lambda_j} \rho_i \frac{(\mathbf{b}_j^H \mathbf{h}_{i,j})^H}{\|\mathbf{b}_j^H \mathbf{h}_{i,j}\|^2}. \quad (14)$$

Under the transmit power constraint $|\alpha_i|^2 \leq P_S$,

$$\left| \sqrt{\lambda_j} \rho_i \frac{(\mathbf{b}_j^H \mathbf{h}_{i,j})^H}{\|\mathbf{b}_j^H \mathbf{h}_{i,j}\|^2} \right|^2 \leq P_S. \quad (15)$$

Given that $\mathbf{b}_j^H \mathbf{h}_{i,j} \in \mathbb{C}$, it can be derived that

$$\left| (\mathbf{b}_j^H \mathbf{h}_{i,j})^H \right|^2 = \left| (\mathbf{b}_j^H \mathbf{h}_{i,j}) \right|^2 = \left\| \mathbf{b}_j^H \mathbf{h}_{i,j} \right\|^2. \quad (16)$$

Therefore, combining (14) and (15), the normalization factor can be expressed and bounded as

$$\lambda_j = \frac{\alpha_i^2}{\rho_i^2} \left\| \mathbf{b}_j^H \mathbf{h}_{i,j} \right\|^2 \leq \frac{P_S \left\| \mathbf{b}_j^H \mathbf{h}_{i,j} \right\|^2}{\rho_i^2}. \quad (17)$$

To fully utilize the transmission power and suppress noise, the normalization factor should be set to its upper bound as

$$\lambda_j = \min_{s_i \in \mathcal{S}^{[e]}} \frac{P_S \left\| \mathbf{b}_j^H \mathbf{h}_{i,j} \right\|^2}{\rho_i^2}. \quad (18)$$

Thus, the MSE is given by

$$\text{MSE}(\hat{p}_j, p_j; \mathcal{S}^{[e]}, \mathbf{b}_j) = \frac{\|\mathbf{b}_j\|^2 \sigma^2}{\lambda_j} \quad (19a)$$

$$= \frac{\sigma^2}{P_S} \max_{s_i \in \mathcal{S}^{[e]}} \rho_i^2 \frac{\|\mathbf{b}_j\|^2}{\left\| \mathbf{b}_j^H \mathbf{h}_{i,j} \right\|^2}. \quad (19b)$$

C. Problem Formulation

In FL, a critical factor that affects the performance of the global model $m^{[e]}$ is the volume of training data $\mathbf{D}^{[e]} = (|d_1^{[e]}|, |d_2^{[e]}|, \dots, |d_S^{[e]}|)^T$ used to train local models $\mathcal{M}^{[e]} = \{m_1^{[e]}, m_2^{[e]}, \dots, m_S^{[e]}\}$. Thus, the ideal strategy is to aggregate all local models in each epoch e . However, this conflicts with the principle of AFL, which relaxes the aggregation constraint of SFL and this is difficult to achieve due to the limitations caused by AirComp, as presented in Section II-B. Furthermore, aggregation errors may lead to a notable drop in prediction accuracy [37]. Since not all sensing UAVs \mathcal{S} can be connected to communication UAVs \mathcal{C} , the optimization target is to maximize the total training data volume $\|\mathbf{L}^{[e]} \mathbf{D}\|_1$ utilized by aggregated local models $\mathcal{M}_{\text{upload}}^{[e]}$. This is achieved by jointly adjusting the linkage scheme $\mathbf{L}^{[e]}$ and the beamforming vectors \mathbf{b}_j at epoch e , subject to MSE and power constraints from (19) and (11) as

$$\underset{\mathbf{L}^{[e]} \in \mathbb{R}^{\mathcal{C} \times \mathcal{S}}, \mathbf{b}_j \in \mathbb{C}^A}{\text{maximize}} \left\| \mathbf{L}^{[e]} \mathbf{D} \right\|_1 \quad (20a)$$

$$\text{s.t.} \quad \frac{\sigma^2}{P_S} \max_{s_i \in \mathcal{S}^{[e]}} \rho_i^2 \frac{\|\mathbf{b}_j\|^2}{\left\| \mathbf{b}_j^H \mathbf{h}_{i,j} \right\|^2} \leq \gamma, \quad \forall j, \quad (20b)$$

$$\|\mathbf{b}_j\|^2 \leq P_N, \quad c_j \in \mathcal{C} \quad (20c)$$

where the operation $\|\cdot\|_1$ denotes the Manhattan norm, i.e., the sum of the absolute value of all elements in a matrix or vector, and γ refers to the MSE threshold of AirComp transmission. Solving the mixed combinatorial optimization problem (20) is challenging due to the combinatorial objective function $\|\mathbf{L}^{[e]} \mathbf{D}\|_1$ and the nonconvex MSE constraint.

The transceiver design proposed here, which utilizes AirComp, assumes the availability of perfect channel state information (CSI) [38]. However, obtaining and transmitting CSI feedback can introduce considerable overhead [39]. To address this issue, an alternative method involves performing the transceiver design at the communication UAVs \mathcal{C} by solving problem (20) and computing the result in equation (13), which only requires the CSI at \mathcal{C} [40]. Once the transmitter scalars are determined, \mathcal{C} can transmit feedback to sensing UAVs \mathcal{S} by sending the corresponding scalar value α_i . To estimate the CSI at \mathcal{C} , channel training can be performed by transmitting pilot sequences from \mathcal{S} [41]. The feedback process can then be

handled using either unquantized analog feedback or quantized digital feedback [42].

III. SOLUTIONS FOR LINKAGE SCHEMES AND BEAMFORMING VECTORS

To solve problem (20), we decompose it into two subproblems: one for the linkage scheme $\mathbf{L}^{[e]}$ and one for the beamforming vector \mathbf{b}_j , denoted as \mathcal{P}_1 and \mathcal{P}_2 , respectively.

The first subproblem aims to determine the optimal linkage matrix $\mathbf{L}^{[e]} \in \{0,1\}^{C \times S}$ that maximizes the total amount of training data utilized for aggregation. Each element $l_{i,j}^{[e]} \in \{0,1\}$ indicates whether sensing UAV s_i uploads its local model to communication UAV c_j at epoch e . Accordingly, the uploading plan of sensing UAVs is defined as $\mathbf{v}^{[e]} = \sum_{j=1}^C \mathbf{L}_{j,:}^{[e]}$, which represents whether each sensing UAV participates in the current aggregation epoch.

$$\mathcal{P}_1 : \underset{\mathbf{L}^{[e]} \in \{0,1\}^{C \times S}}{\text{maximize}} \|\mathbf{L}^{[e]} \mathbf{D}\|_1 \quad (21a)$$

$$\text{s.t. } \mathbf{v}^{[e]} = \sum_{j=1}^C \mathbf{L}_{j,:}^{[e]} \quad (21b)$$

$$\mathbf{v}^{[e]} \mathbf{v}_k^{\top} < \|\mathbf{v}_k\|_1, \quad \forall \mathbf{v}_k \in \mathcal{V}_{\text{NF}}^{[e]} \quad (21c)$$

$$\mathbf{L}_{j,:}^{[e]} \mathbf{l}_k^{\top} \leq \|\mathbf{l}_k\|_1, \quad \forall c_j \in \mathcal{C}, \mathbf{l}_k \in \mathcal{L}_{\text{NF},j}^{[e]} \quad (21d)$$

Here, $\mathcal{V}_{\text{NF}}^{[e]}$ denotes the set of previously identified non-feasible uploading plans, while $\mathcal{L}_{\text{NF},j}^{[e]}$ represents the set of non-feasible transmission plans for communication UAV c_j . These sets are not optimization variables but are progressively constructed during the branch-and-bound procedure.

Specifically, if a candidate linkage matrix $\mathbf{L}^{[e]}$ yields an infeasible beamforming solution in subproblem \mathcal{P}_2 , the corresponding uploading plan $\mathbf{v}^{[e]}$ or transmission plan $\mathbf{L}_{j,:}^{[e]}$ is recorded in $\mathcal{V}_{\text{NF}}^{[e]}$ or $\mathcal{L}_{\text{NF},j}^{[e]}$, respectively. The additional constraints above prevent repeated exploration of these infeasible plans, thereby pruning the search space efficiently.

With binary decision variables and linear constraints, subproblem \mathcal{P}_1 becomes a 0–1 integer linear programming (ILP) problem. It is solved using a branch-and-bound algorithm that systematically explores feasible linkage schemes in descending order of the objective value while excluding infeasible branches identified by \mathcal{P}_2 .

The second subproblem is formulated as

$$\mathcal{P}_2 : \underset{\mathbf{b}_j \in \mathbb{C}^A}{\text{minimize}} \|\mathbf{b}_j\|^2 \quad (22a)$$

$$\text{s.t. } \frac{\sigma^2}{P_S} \max_{s_i \in \mathcal{S}^{[e]}} l_{i,j} \rho_i^2 \frac{\|\mathbf{b}_j\|^2}{\|\mathbf{b}_j^{\text{H}} \mathbf{h}_{i,j}\|^2} \leq \gamma, \quad \forall j, \quad (22b)$$

$$\|\mathbf{b}_j\|^2 \leq P_N, c_j \in \mathcal{C} \quad (22c)$$

to find the beamforming vectors \mathbf{b}_j to support the linkage scheme $\mathbf{L}^{[e]}$ obtained at subproblem (21). To simplify the

subproblem, define a new threshold $\gamma_i = \frac{\gamma P_S}{\sigma^2 \rho_i^2}$, and rewrite subproblem \mathcal{P}_2 as

$$\mathcal{P}_2 : \underset{\mathbf{b}_j \in \mathbb{C}^A}{\text{minimize}} \|\mathbf{b}_j\|^2 \quad (23a)$$

$$\text{s.t. } \max_{s_i \in \mathcal{S}^{[e]}} l_{i,j} \frac{\|\mathbf{b}_j\|^2}{\|\mathbf{b}_j^{\text{H}} \mathbf{h}_{i,j}\|^2} \leq \gamma_i, \quad \forall j, \quad (23b)$$

$$\|\mathbf{b}_j\|^2 \leq P_N, c_j \in \mathcal{C}. \quad (23c)$$

If a beamforming vector \mathbf{b}_j satisfies constraint (23b), then its scaled version $\beta \mathbf{b}_j$ also satisfies the constraint, because

$$\frac{\|\beta \mathbf{b}_j\|^2}{\|\beta \mathbf{b}_j^{\text{H}} \mathbf{h}_{i,j}\|^2} = \frac{\beta^2 \|\mathbf{b}_j\|^2}{\beta^2 \|\mathbf{b}_j^{\text{H}} \mathbf{h}_{i,j}\|^2} = \frac{\|\mathbf{b}_j\|^2}{\|\mathbf{b}_j^{\text{H}} \mathbf{h}_{i,j}\|^2}. \quad (24)$$

Therefore, there must exist $\beta > 0$ such that $\|\beta \mathbf{b}_j\|^2 \leq P_N$ and thus the solution of subproblem (21) must satisfy the transformed subproblem

$$\mathcal{P}'_2 : \underset{\mathbf{b}_j \in \mathbb{C}^A}{\text{minimize}} \|\mathbf{b}_j\|^2 \quad (25a)$$

$$\text{s.t. } \max_{s_i \in \mathcal{S}^{[e]}} l_{i,j} \frac{P_N}{\|\mathbf{b}_j^{\text{H}} \mathbf{h}_{i,j}\|^2} \leq \gamma_i, \quad \forall j, \quad (25b)$$

$$\|\mathbf{b}_j\|^2 \leq P_N, c_j \in \mathcal{C}. \quad (25c)$$

Conversely, if there exists \mathbf{b}_j that satisfies the transformed subproblem (25), then

$$\gamma_i \geq \frac{l_{i,j} P_N}{\|\mathbf{b}_j^{\text{H}} \mathbf{h}_{i,j}\|^2} \geq \frac{l_{i,j} \|\mathbf{b}_j\|^2}{\|\mathbf{b}_j^{\text{H}} \mathbf{h}_{i,j}\|^2}, \quad \forall i \quad (26)$$

and therefore the solution \mathbf{b}_j of the subproblem (25) must satisfy the constraints of subproblem (21) as well. Since the solutions to subproblem (21) and (25) satisfy the constraints of each other, these two subproblems are equivalent. Then the subproblem (25) can be equivalently written as

$$\mathcal{P}_2 : \underset{\mathbf{b}_j \in \mathbb{C}^A}{\text{minimize}} \|\mathbf{b}_j\|^2 \quad (27a)$$

$$\text{s.t. } \|\mathbf{b}_j^{\text{H}} \mathbf{h}_{i,j}\|^2 \geq \tilde{\gamma}_i, \quad \forall i, j \quad (27b)$$

$$\|\mathbf{b}_j\|^2 \leq P_N, c_j \in \mathcal{C} \quad (27c)$$

with a new threshold $\tilde{\gamma}_i = \frac{l_{i,j} P_N}{\gamma_i}$. Here, the constraint (27b) is nonconvex, while (27c) is convex. Up to this point, the problem (20) can be solved by alternately solving the two subproblems (21) and (27), as shown in Algorithm 2.

To solve the subproblem (27), the main difficulty is the nonconvex constraint (27b), which indicates that the inner product $\mathbf{b}_j^{\text{H}} \mathbf{h}_{i,j}$ is a complex scalar and its squared modulus is greater than a threshold $\tilde{\gamma}_i$. As a complex scalar, it can be rotated on the complex plane by

$$\theta_i = -\arg(\mathbf{b}_j^{\text{H}} \mathbf{h}_{i,j}) \quad (28)$$

and aligned with the real axis as

$$\text{Re}(\mathbf{b}_j^{\text{H}} \mathbf{h}_{i,j} \cdot e^{i\theta_i}) = \|\mathbf{b}_j^{\text{H}} \mathbf{h}_{i,j} \cdot e^{i\theta_i}\| = \|\mathbf{b}_j^{\text{H}} \mathbf{h}_{i,j}\|. \quad (29)$$

Algorithm 2 Algorithm for Linkage Scheme and Beamforming Vector

Initialize non-feasible sets $\mathcal{V}_{\text{NF}}^{[e]}$ and $\mathcal{L}_{\text{NF},j}^{[e]}$
while exists uploading plan $\mathbf{v}^{[e]}$ by solving \mathcal{P}_1 **do**
 while exists transmitting plan $\mathcal{L}^{[e]}$ by solving \mathcal{P}_1 **do**
 if Algorithm 3 solves for beamforming vector $\mathbf{b}_j, \forall c_j \in \mathcal{C}$ by solving \mathcal{P}_2 **then**
 Return $\mathcal{L}^{[e]}$ and \mathbf{b}_j
 else
 Add $\mathcal{L}^{[e]}$ to $\mathcal{L}_{\text{NF},j}^{[e]}$
 end if
 end while
 Add $\mathbf{v}^{[e]}$ to $\mathcal{V}_{\text{NF}}^{[e]}$
end while

In other words, if there exists a beamforming vector \mathbf{b}_j that satisfies the constraint (27b), there must exist $\mathbf{b}'_j = \mathbf{b}_j \cdot e^{i\theta_i}$ that satisfies the constraint as

$$\text{Re}(\mathbf{b}'_j{}^H \mathbf{h}_{i,j}) = \text{Re}(\mathbf{b}_j^H \mathbf{h}_{i,j} \cdot e^{i\theta_i}) = \left\| \mathbf{b}_j^H \mathbf{h}_{i,j} \right\| \geq \sqrt{\tilde{\gamma}_i}. \quad (30)$$

Therefore, the subproblem (27) can be rewritten as

$$\mathcal{P}_2 : \underset{\mathbf{b}_j \in \mathbb{C}^A, \theta \in \mathbb{R}^S}{\text{minimize}} \left\| \mathbf{b}_j \right\|^2 \quad (31a)$$

$$\text{s.t. } \text{Re}(\mathbf{b}_j^H \mathbf{h}_{i,j} \cdot e^{i\theta_i}) \geq \sqrt{\tilde{\gamma}_i}, \quad \forall i, j \quad (31b)$$

$$\boldsymbol{\theta} = (\theta_1, \theta_2, \dots, \theta_S) \quad (31c)$$

$$\left\| \mathbf{b}_j \right\|^2 \leq P_N, c_j \in \mathcal{C} \quad (31d)$$

where the auxiliary angle vector $\boldsymbol{\theta}$ includes the auxiliary angles θ_i to help solve the subproblem.

In this way, the constraint (31b) is no longer nonconvex if θ_i is regarded as a constant whereas when \mathbf{b}_j is regarded as a constant θ_i can be easily calculated by (28) as well. Therefore, an alternating optimization algorithm can be used to solve the subproblem (31) in two steps, including

$$\mathcal{P}_{2-1} : \underset{\mathbf{b}_j \in \mathbb{C}^A}{\text{minimize}} \left\| \mathbf{b}_j \right\|^2 \quad (32a)$$

$$\text{s.t. } \text{Re}(\mathbf{b}_j^H \mathbf{h}_{i,j} \cdot e^{i\theta_i}) \geq \sqrt{\tilde{\gamma}_i}, \quad \forall i, j \quad (32b)$$

and

$$\mathcal{P}_{2-2} : \underset{\boldsymbol{\theta} \in \mathbb{R}^S}{\text{minimize}} \text{Re}(\mathbf{b}_j^H \mathbf{h}_{i,j} \cdot e^{i\theta_i}) \quad \forall i, j \quad (33a)$$

$$\text{s.t. } \boldsymbol{\theta} = (\theta_1, \theta_2, \dots, \theta_S). \quad (33b)$$

The first step, \mathcal{P}_{2-1} , is convex and can be solved using the interior-point method or sequential quadratic programming. The second step, \mathcal{P}_{2-2} , is convex as well and can be calculated by (28) directly.

The sequence $\{(\mathbf{b}_j^{[t]}, \boldsymbol{\theta}^{[t]})\}$ generated by iteratively solving \mathcal{P}_{2-1} and \mathcal{P}_{2-2} yields a non-increasing and convergent sequence of objective values $\left\{ \left\| \mathbf{b}_j^{[t]} \right\|^2 \right\}$ as shown in the following proposition.

Proposition 1: Let $\mathbf{b}_j^{[t]}$ as the result of \mathcal{P}_{2-1} at the iteration t of the proposed algorithm and $\boldsymbol{\theta}^{[t]}$ as the result of \mathcal{P}_{2-2} .

Since the target of \mathcal{P}_{2-2} is to maximize $\text{Re}(\mathbf{b}_j^H \mathbf{h}_{i,j} \cdot e^{i\theta_i}), \forall i, j$, it follows that

$$\text{Re}(\mathbf{b}_j^{[t]}{}^H \mathbf{h}_{i,j} \cdot e^{i\theta_i^{[t]}}) \geq \text{Re}(\mathbf{b}_j^{[t]}{}^H \mathbf{h}_{i,j} \cdot e^{i\theta_i^{[t-1]}}) \geq \sqrt{\tilde{\gamma}_i}. \quad (34)$$

Hence, there exists $\boldsymbol{\mu}^{[t]} = \left\{ \mu_i^{[t]} \mid \mu_i^{[t]} \in (0, 1] \right\}$ so that

$$\text{Re}(\mu_i^{[t]} \mathbf{b}_j^{[t]}{}^H \mathbf{h}_{i,j} \cdot e^{i\theta_i^{[t]}}) = \text{Re}(\mathbf{b}_j^{[t]}{}^H \mathbf{h}_{i,j} \cdot e^{i\theta_i^{[t-1]}}) \geq \sqrt{\tilde{\gamma}_i}. \quad (35)$$

In this way, at the iteration $t+1$ of the proposed algorithm, the subproblem \mathcal{P}_{2-1} can be obtained with the target value less than or equal to that of the last iteration as

$$\left\| \mathbf{b}_j^{[t+1]} \right\|^2 \leq \left\| \mu_{\max}^{[t]} \mathbf{b}_j^{[t]} \right\|^2 \leq \left\| \mathbf{b}_j^{[t]} \right\|^2 \quad (36)$$

where $\mu_{\max}^{[t]} = \max \boldsymbol{\mu}^{[t]}$. As a result, in the proposed algorithm, when iteratively solving subproblems \mathcal{P}_{2-1} and \mathcal{P}_{2-2} , the sequence $\left\{ \left\| \mathbf{b}_j^{[1]} \right\|^2, \dots, \left\| \mathbf{b}_j^{[t]} \right\|^2 \right\}$ of the target values of problem \mathcal{P}_2 is non-increasing. Assuming \mathbf{b}_j^* is the local minimum solution to problem (31) with $\left\| \mathbf{b}_j^* \right\|^2 > 0$, the sequence must converge to a value greater than or equal to $\left\| \mathbf{b}_j^* \right\|^2$, representing the local optimum. To stop the iteration, the ending criterion is set as

$$\frac{\left\| \mathbf{b}_j^{[t-1]} \right\|^2 - \left\| \mathbf{b}_j^{[t]} \right\|^2}{\left\| \mathbf{b}_j^{[t]} \right\|^2} \leq \epsilon \quad (37)$$

where $\epsilon > 0$ is the threshold. The details of the proposed alternating optimization algorithm are illustrated in Algorithm 3.

The computational complexity of the two subproblems should be analysed separately. Consider one communication UAV $c_j \in \mathcal{C}$ with A antennas connects with all sensing UAVs \mathcal{S} whose number is S , the subproblems \mathcal{P}_{2-1} can be regarded as having $2A$ real variables in \mathbf{b}_j and S constraints. When solving \mathcal{P}_{2-1} with an interior-point method, the complexity is approximately $\mathcal{O}((2A+S)^{1.5}S)$ iterations, with each iteration costing $\mathcal{O}(2AS + (2A)^2)$. Thus, the total complexity is

$$\mathcal{O}((2A+S)^{1.5}S) \times \mathcal{O}(2AS + (2A)^2) = \mathcal{O}(AS^{3.5}) \quad (38)$$

Algorithm 3 Alternating Optimization Algorithm for Beamforming Vectors

Initialize auxiliary angle vector $\boldsymbol{\theta}^{[0]}$
 Get beamforming vector $\mathbf{b}^{[0]}$ through solving \mathcal{P}_{2-1}
while $\mathbf{b}^{[t]}$ does not satisfy condition (37) **do**
 Update auxiliary angle vector $\boldsymbol{\theta}^{[t]}$ through solving \mathcal{P}_{2-2}
 Update beamforming vector $\mathbf{b}^{[t]}$ through solving \mathcal{P}_{2-1}
end while
if $\mathbf{b}^{[t]}$ satisfies constraint (27b) **then**
 Return the optimized beamforming vector $\mathbf{b}^{[t]}$
else
 No feasible beamforming vector exists
end if

where $S \gg A$. Then the subproblem \mathcal{P}_{2-2} requires $\mathcal{O}(A)$ to calculate $\mathbf{b}_j^H \mathbf{h}_{i,j}$, $\mathcal{O}(1)$ to extract phase $\arg(\mathbf{b}_j^H \mathbf{h}_{i,j})$, and finally $\mathcal{O}(1)$ to set $\theta_i = -\arg(\mathbf{b}_j^H \mathbf{h}_{i,j})$. Therefore, with S elements in $\boldsymbol{\theta}$, the complexity of \mathcal{P}_{2-2} is $\mathcal{O}(AS)$. In this way, the total computation complexity of the proposed algorithm is

$$\mathcal{O}(AS^{3.5}) + \mathcal{O}(AS) = \mathcal{O}(AS^{3.5} + AS) \approx \mathcal{O}(AS^{3.5}). \quad (39)$$

By comparison, the branch-and-bound algorithm for the convex-argument-cut-based relaxation problem (ACR-BB) that outputs the global optimal solutions has a complexity of $\mathcal{O}(A^3 S^{3.5})$ [43]. The simulation results in V-B show that the proposed algorithm achieves better performance and lower computational complexity than the ACR-BB algorithm through multiple random initializations.

According to the property

$$\left\| \mathbf{b}_j^H \mathbf{h}_{i,j} \right\|^2 \leq \|\mathbf{b}_j\|^2 \cdot \|\mathbf{h}_{i,j}\|^2 \quad (40)$$

and constraint (27b), the potential minimum value of objective function $\|\mathbf{b}_j\|^2$ is derived as

$$\|\mathbf{b}_j\|^2 \cdot \|\mathbf{h}_{i,j}\|^2 \geq \left\| \mathbf{b}_j^H \mathbf{h}_{i,j} \right\|^2 \geq \tilde{\gamma}_i \Rightarrow \|\mathbf{b}_j\|^2 \geq \frac{\tilde{\gamma}_i}{\|\mathbf{h}_{i,j}\|^2} \quad (41)$$

and the potential maximum value is P_C . Based on the terminal condition (37), the longest sequence of $\left\{ \left\| \mathbf{b}_j^{[t]} \right\|^2 \right\}$ satisfies the following relationship:

$$\mathbf{b}_j^{[t]} = \frac{1}{1 + \epsilon} \mathbf{b}_j^{[t-1]}, \forall t \neq T_{\max} \quad (42)$$

which can be represented as

$$P_C \left(\frac{1}{1 + \epsilon} \right)^{T_{\max} - 1} < \frac{\tilde{\gamma}_i}{\|\mathbf{h}_{i,j}\|^2} \leq P_C \left(\frac{1}{1 + \epsilon} \right)^{T_{\max}}. \quad (43)$$

Here, T_{\max} is the sequence length, representing the iteration complexity, and is given by

$$T_{\max} = \left\lceil \log_{1+\epsilon} \left(\frac{P_C \|\mathbf{h}_{i,j}\|^2}{\tilde{\gamma}_i} \right) \right\rceil. \quad (44)$$

IV. SELF-ADAPTIVE AGGREGATION SCHEME FOR MODEL STALENESS

The first difficulty in mitigating model staleness is the limitation of information availability caused by AirComp in the proposed AFL framework. The conventional process of aggregating the local models [29] can be described as

$$\underset{\tilde{\rho}_i \in [0,1]}{\text{minimize}} \mathcal{J} \left(m^{[e]} \right) \quad (45a)$$

$$\text{s.t. } m^{[e]} = \left(1 - \sum_{s_i \in \mathcal{S}^{[e]}} \tilde{\rho}_i \right) m^{[e-1]} + \sum_{s_i \in \mathcal{S}^{[e]}} \tilde{\rho}_i m_i^{[e]} \quad (45b)$$

$$\sum_{s_i \in \mathcal{S}^{[e]}} \tilde{\rho}_i \leq 1 \quad (45c)$$

where $\mathcal{J}(\cdot)$ is the loss function of the global model $m^{[e]}$ and the scalar $\tilde{\rho}_i$ is the weight of the local model $m_i^{[e]}$ in the global model $m^{[e]}$. By controlling these scalars, PSs in conventional

FL frameworks can mitigate model staleness in less valuable local models and enhance the contribution of more valuable ones. However, this is difficult to achieve in our proposed AFL framework because communication UAVs receive only superposed models due to AirComp, rather than individual models. Therefore, the process of mitigating model staleness can only be implemented on the sensing UAVs, but they have limited access to information of other local models.

The second difficulty is that simple aggregation schemes may lead to the loss of useful knowledge while reducing model staleness. Some related studies [29], [44] have proposed simple aggregation schemes to mitigate model staleness in FL as

$$\underset{\tau_e \in \mathbb{N}^+}{\text{minimize}} \mathcal{J} \left(m^{[e]} \right) \quad (46a)$$

$$\text{s.t. } m^{[e]} = (1 - \rho) \cdot m^{[e-1]} + \frac{1}{\psi} \sum_{s_i \in \mathcal{S}^{[e]}} \rho_i \cdot m_i^{[e]} \quad (46b)$$

$$\rho = \frac{1}{\psi} \sum_{s_i \in \mathcal{S}^{[e]}} \rho_i \quad (46c)$$

$$\rho_i = \sum_{k=e'_i}^e \left| d_i^{[k]} \right| \cdot \frac{e'_i}{e} \quad (46d)$$

$$\psi = \sum_{k=e'_i}^e \left| d_i^{[k]} \right| \quad (46e)$$

$$e - e'_i > \tau_e, \forall s_i \in \mathcal{S}^{[e]} \quad (46f)$$

which can be implemented in our proposed AFL framework. Here τ_e is the threshold on the staleness delay to drop local models with high staleness and other variables are defined in Section II and ρ is the sum of the weight ρ_i applied to local models. However, simply applying scalars to local models or dropping those that have been trained for a long time is a blunt solution. While it reduces the influence of highly stale local models and prevents bias in the global model, it also diminishes the impact of the knowledge those models have acquired. Typically, models with high staleness have been trained for longer periods and with larger data volumes than those with low staleness, suggesting that these models may have learned more and could perform better in certain situations. Therefore, preserving parts of these models may contribute to improving global models, despite their staleness.

To address the above two challenges while reducing model staleness, we propose a self-adaptive aggregation scheme that enables sensing UAVs to adaptively modify their own local models. Specifically, sensing UAVs modify particular layers of their local models if those layers exhibit low cosine similarity with the corresponding layers of the latest global model. Cosine similarity is used to determine which parts of a model should be preserved and which should be modified, thereby measuring the staleness of local models. Assuming independent and identically distributed (IID) data [45], the gradients of all local models can be considered to point in similar directions. Moreover, the more data a deep learning model has used for training, the closer it is to the optimal target shared by all local and global models [46], [47]. Therefore, the global model, which contains the most comprehensive knowledge, serves as the reference for comparison with local models. The higher the similarity between a local model and

the global model, the less staleness it contains and the more valuable it is for inclusion in the new global model.

In the third step in Fig. 3, previously denoted by $f(\cdot)$ before in Section II, at the e^{th} global epoch, sensing UAV i first receives the latest global model $m^{[e-1]}$ if it is selected to upload its local model $m_i^{[e]}$ in this epoch. Then it compares its local model $m_i^{[e]}$ to the global model $m^{[e-1]}$ layer by layer by calculating the cosine similarity as

$$\text{cos-sim}(w_i^k, w^k) = \frac{w_i^k \cdot w^k}{\|w_i^k\| \cdot \|w^k\|} \quad (47)$$

where $w_i^k \in m_i^{[e]}$ and $w^k \in m^{[e-1]}$ denote the k^{th} layers of the local model $m_i^{[e]}$ and global model $m^{[e-1]}$, respectively. In this way, the aggregation scheme in (4) appears as

$$m^{[e]} = (1 - \rho) \cdot m^{[e-1]} + \frac{1}{\psi} \sum_{s_i \in \mathcal{S}^{[e]}} \rho_i \cdot \widehat{m}_i^{[e]} \quad (48)$$

where the local model $\widehat{m}_i^{[e]}$ is self-adapted. To clarify, the aggregation scheme (4) and (48) are the same except for the simplified scalar $\rho = \frac{1}{\psi} \sum_{s_i \in \mathcal{S}^{[e]}} \rho_i$. This model consists of layers \widehat{w}_i^k selected from either the original local model $m_i^{[e]}$ or the latest global model $m^{[e-1]}$ as

$$\widehat{w}_i^k = \mathbb{I}(\mathcal{R}_i^k \geq \tau_{\text{cos}}) w_i^k + \mathbb{I}(\mathcal{R}_i^k < \tau_{\text{cos}}) w^k \quad (49)$$

where the constant \mathcal{R}_i^k is calculated as

$$\mathcal{R}_i^k = \text{cos-sim}(w_i^k, w^k). \quad (50)$$

Here, the indicator function $\mathbb{I}(\cdot) = 1$ when the condition is true and $\mathbb{I}(\cdot) = 0$ when the condition is false. Specifically, if the cosine similarity $\text{cos-sim}(w_i^k, w^k)$ exceeds a threshold τ_{cos} , the k^{th} layer of local model $m_i^{[e]}$ is preserved as w_i^k , otherwise, it is replaced by the corresponding layer of the global model w^k . This process is illustrated in Algorithm 4.

The threshold τ_{cos} should be set as an adjustable value that increases with the global epoch e . This is because at the initial stage of AFL, the global model is not well-trained and the training of local models leads to large divergence among them. Therefore, the cosine similarity between local models and the global model is typically low, and the threshold τ_{cos} should be set to a low value to preserve more knowledge learned by the local models. As training proceeds, the global model is gradually optimized and local models are updated accordingly, resulting in higher cosine similarity. Thus, the threshold τ_{cos} can be increased to replace more layers in the local models with those from the global model, thereby reducing model staleness. For example, in the simulations in Section V, τ_{cos} is set to 0.5 at the first global epoch and gradually increased to 0.9 at the last global epoch.

V. SIMULATIONS

In this section, we conduct simulations to compare the proposed alternating optimization algorithm for beamforming vectors with a state-of-the-art (SOTA) algorithm that derives the globally optimal solutions. We then conduct experiments to evaluate the performance of the proposed self-adaptive aggregation scheme in both SFL and AFL, with a case study

Algorithm 4 Self-Adaptive Aggregation Scheme

Sensing UAVs $\mathcal{S}^{[e]}$ receive the latest global model $m^{[e-1]}$ from communication UAVs \mathcal{C}
for each local model $m_i^{[e]} \in \mathcal{M}_{\text{upload}}^{[e]}$ **do**
 for each layer of model $w_i^k \in m_i^{[e]}$ **do**
 if $\text{cos-sim}(w_i^k, w^k) \leq \tau_{\text{cos}}$ **then**
 Replace layer in local model $m_i^{[e]}$: $w_i^k := w^k$
 end if
 end for
Sensing UAV s_i obtains the self-adapted local model $\widehat{m}_i^{[e]}$
end for
 $\mathcal{S}^{[e]}$ upload self-adapted local models $\mathcal{M}'_{\text{upload}}^{[e]} = \{\widehat{m}_i^{[e]} \mid s_i \in \mathcal{S}^{[e]}\}$ to \mathcal{C}

on how the model staleness affects the performance of the AFL model and how the proposed aggregation scheme can compensate for this issue. Finally, we apply the proposed AFL framework to UAV-related distributed deep learning tasks to demonstrate its effectiveness.

A. Simulation Setup

UAV Swarm and Communication Model. In the simulations, we consider a UAV swarm in which the communication UAVs equipped with $A = 5$ antennas and the sensing UAVs equipped with a single antenna. In a region of $20 \times 20 \times 20$ cubic kilometres, the UAV trajectories are determined by the Olfati-Saber algorithm [10], which is well known for distributed control of multi-agent systems. This mobility model ensures realistic swarm coordination and dynamic spatial configurations during the learning process. However, it may introduce Doppler shifts and time-varying propagation delays, which are assumed to be compensated for through standard synchronization techniques at the physical layer in simulations. Furthermore, the wireless channels between UAVs are modelled using the Rician channel model since the UAVs are considered to communicate under line-of-sight (LoS) conditions [44] and channel coefficients are updated according to instantaneous UAV positions at each epoch. Some simulation parameters of the simulations are listed in Table I.

Deep Learning. To verify the effectiveness and generality of our proposed self-adaptive aggregation scheme in AFL, we conduct experiments on two datasets: the Modified National Institute of Standards and Technology database (MNIST) [48] and VisDrone2019 dataset [49]. The MNIST dataset is a widely used collection of handwritten digit images, consisting of 60,000 training examples and 10,000 testing examples, and serves as a standard benchmark in the previous FL studies for its simplicity and representativeness. The VisDrone2019 dataset consists of 288 video clips comprising 261,908 frames and 10,209 static images, which were collected using various drone platforms in different scenarios, weather conditions, and lighting conditions, covering a wide range of aspects including location, environment, objects, and density. This dataset is used to evaluate the performance of the proposed AFL framework in UAV-related distributed deep learning tasks. In terms of training process, since the MNIST dataset

TABLE I
KEY SIMULATION PARAMETERS

Parameters	Values
Number of sensing UAVs	1-20
Number of communication UAVs	1-10
Local training epoch	1
Global training epoch	30
Global epoch duration	120 seconds
Local training time (including data-sensing time)	90 seconds per epoch
Transmission time	30 seconds
Waiting time of retransmission for SFL	30 seconds
Volume of training data	11.2-22.4 GB per epoch per UAV
Number of parameters of a deep learning model	1.88×10^9
Transmission power of UAV (reference)	10 Watts (W)

involves 10 classes, it is split into IID data [45] based on the number of UAVs in each simulation. However, the variation of the VisDrone2019 dataset is too complex to be divided in an independent and identically distributed manner, thus it is considered as non-IID data [25]. We set the local training epoch to 1 to minimize the risk of early divergence in the global model by ensuring more frequent aggregation of updates, which helps maintain model consistency and prevent biases caused by non-IID data across nodes. In the subsequent simulation results, SFL and AFL refer to synchronous and asynchronous federated learning, respectively, and 'Adapt' refers to the proposed self-adaptive aggregation scheme.

B. Performance of Proposed Alternating Optimization Algorithm for Beamforming Vectors

Fig. 5 shows the performance of the proposed alternating optimization algorithm for solving beamforming vectors. To obtain the optimal value, we use the ACR-BB algorithm to determine the lower and upper bounds of the global optimal solutions. As described in [43], the upper bound is the actual solution derived by the ACR-BB algorithm and the lower bound is the potential value that the solution may achieve. Thus, the global optimal solution lies within this range.

The vertical axis in the figure represents the ratio of the difference between the solution value and the lower bound value to the lower bound value. As shown in the figure, the solutions obtained by the proposed algorithm consistently lie between the upper and lower bounds of the optimal solution across all scenarios, indicating that the solution quality of the proposed algorithm is close to the optimal solution. When there is only one transmitter, the original problem can be equivalently transformed into a nonconvex problem, allowing the optimal solution to be directly obtained. However, as the number of transmitters increases, the number of constraints grows, and the problem complexity rises accordingly. Consequently, the solution quality gradually stabilizes near the optimal level.

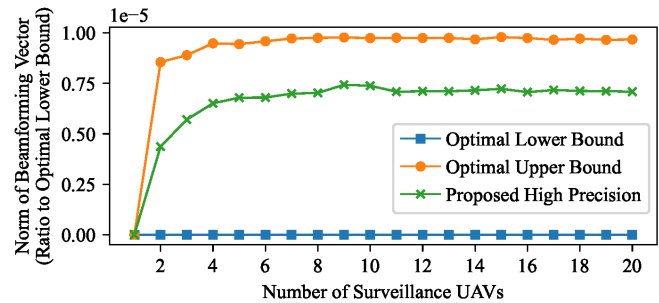


Fig. 5. Average norm of the beamforming vectors over 100 repeated simulations. Because the absolute values of the solutions are too close to be distinguished, the vertical axis is set as the ratio of the difference between the solution value and the lower bound value to the lower bound value.

As previously discussed, the computational complexity of the proposed algorithm is significantly lower than that of the ACR-BB algorithm. Therefore, the proposed algorithm outperforms existing approximation methods in both solution quality and computational efficiency.

Fig. 6 shows the performance of the proposed alternating optimization algorithm for solving beamforming vectors with different numbers of antennas. The reference value is the upper bound derived by the ACR-BB algorithm. Under different MSE thresholds for AirComp, the UAV using the proposed algorithm can use lower energy to transmit the models than that using the ACR-BB algorithm, and that energy decreases as the number of antennas increases, namely the number of optimization variables in the problem, indicating the better performance of our proposed algorithm. Besides, the values derived by the proposed algorithm are very close to each other under different MSE thresholds because in our problem it is the phase of the beamforming vector that influences the distortion of the received signal but not its amplitude. The similarity of these values demonstrates the stability of our algorithm as well.

C. Metrics of Proposed Self-Adaptive Aggregation Scheme

Fig. 7 shows three kinds of resources utilized by the FL frameworks to improve the accuracies of their global models to 90%. Among these resources, *Data* refers to the volume of

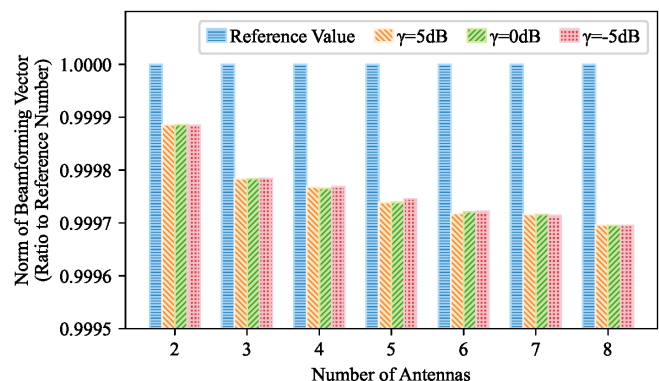


Fig. 6. Average norm of the beamforming vectors of communication UAVs with different numbers of antennas over 100 repeated simulations. The reference value is derived by the ACR-BB algorithm, and it is normalized to 1 because the solution values derived by the proposed algorithm are divided by the reference value of corresponding situations for better comparison.

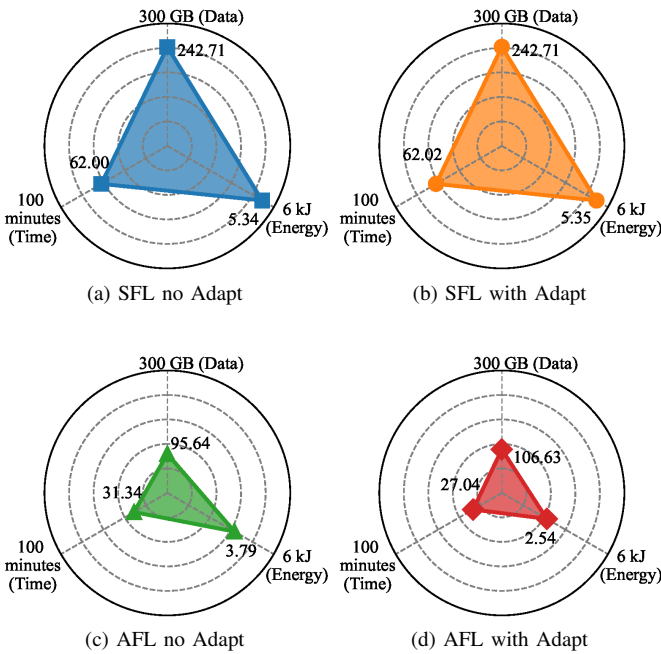


Fig. 7. Consumption of different resources in different FL frameworks.

data transmitted between UAVs, *Time* refers to the total time required to improve the global model, and *Energy* indicates the energy consumption of the communication UAVs to receive local models, which is calculated as the sum of the norm of beamforming vectors for simplification.

In Fig. 7, because AFL collects only a subset of local models instead of all of them and avoids retransmission, it reduces the consumption of all three resources. Besides, the self-adaptive aggregation scheme slightly increases the transmission data in both SFL and AFL because it adds the process of transmitting the latest global models from communication UAVs to sensing UAVs. With this additional cost, global models in AFL requires less time and energy to achieve 90% accuracy, while the global models in SFL show little change.

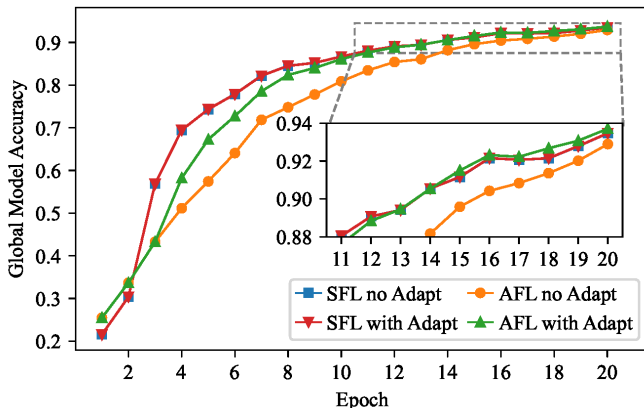


Fig. 8. Comparison between the average accuracies of different FL frameworks over 100 repeated simulations. The results show that our proposed AFL framework can not only compensate for the staleness problem in AFL (total region), but also surpasses SFL by preventing global aggregation from uninformative parts of local models (zoomed-in region).

D. Performance of Self-Adaptive Aggregation Scheme

Fig. 8 shows the performance of the proposed self-adaptive aggregation scheme through the accuracy curves of the global models in different FL frameworks. Since all models almost converge after 20 epochs, Fig. 8 only shows the accuracy curves for the first 20 epochs.

From the main figure, it can be observed that after applying the adaptive aggregation scheme, the model accuracy of SFL shows almost no improvement. This is because SFL requires each global aggregation round to include models from all edge nodes and synchronize the updated global model back to all edge nodes. Consequently, all local models at edge nodes remain fresh (non-stale), making the freshness-aware adaptive aggregation scheme ineffective. In contrast, AFL exhibits a significant improvement in model accuracy, particularly between the third and tenth training rounds. Before the third round, the adaptive aggregation scheme was not applied. This is because, in the initial stages, the model learns limited and highly divergent features, which may lead to low similarity between local and global models due to factors other than staleness (e.g., data heterogeneity), potentially discarding locally useful updates.

The subfigure reveals that as the model approaches convergence, the accuracy across all scenarios becomes comparable. This occurs because, over extended training periods, the global model in all cases accumulates similar amounts of learned knowledge. However, compared to SFL, AFL without adaptive aggregation suffers from noticeable accuracy degradation. This is attributed to model staleness in asynchronous settings, where the global model is skewed by highly stale local updates, causing parameter deviation. Remarkably, AFL with adaptive aggregation not only achieves higher accuracy than its non-adaptive counterpart but also slightly outperforms SFL. This demonstrates that the proposed algorithm effectively mitigates the negative impact of stale models while preserving the deeply learned features from high-staleness models (i.e., those trained for longer durations). As a result, the global model integrates more comprehensive and harder-to-learn knowledge, enhancing its generalization capability.

Fig. 9 shows the training performance of different FL frameworks over time. Compared to Fig. 8, it plots the global model accuracy against average training time instead of the number of epochs, where the duration of each epoch is the mean value of repeated experiments. This representation more accurately reflects the training efficiency of different FL schemes under realistic communication and computation constraints.

Because AFL relaxes the strict synchronization requirement, it allows the server to aggregate available local models at fixed time intervals, whereas SFL must wait until all participating edge nodes complete their local updates and upload their models. As a result, SFL exhibits a longer average epoch duration due to straggling nodes, which leads to slower convergence in terms of absolute time. In contrast, AFL achieves faster accuracy growth by exploiting asynchronous updates and avoiding idle waiting.

Moreover, the proposed self-adaptive aggregation scheme further improves the convergence efficiency of AFL. As in-

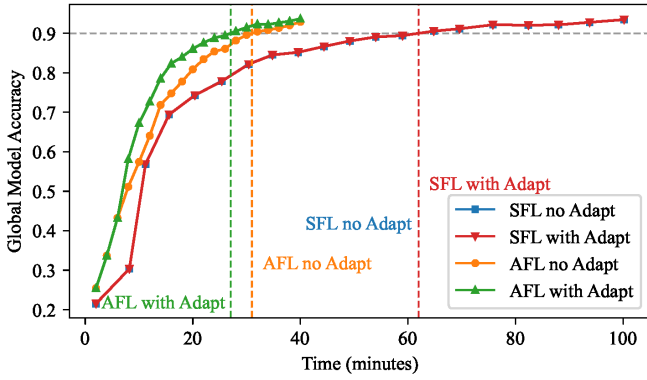


Fig. 9. Comparison between the average accuracies of different FL frameworks over 100 repeated simulations, where the horizontal axis represents the accumulated training time. It can be observed that, although AFL performs more training epochs, its asynchronous nature and the proposed self-adaptive aggregation scheme significantly accelerate convergence in terms of absolute time compared with SFL and AFL without adaptation.

TABLE II
TIME-TO-ACCURACY-THRESHOLD FOR DIFFERENT FL SCHEMES.

Accuracy Threshold	SFL		AFL	
	no Adapt	with Adapt	no Adapt	with Adapt
40%	9.292	9.292	5.321	5.321
50%	10.424	10.424	7.713	6.903
60%	12.299	12.299	10.778	8.400
70%	16.144	16.143	13.526	11.007
80%	27.855	27.878	19.440	14.761
90%	61.998	62.019	30.991	27.014

indicated by the dashed vertical lines corresponding to the accuracy threshold of 90% (indicated by the gray dashed horizontal line), AFL with adaptation reaches the same target accuracy significantly earlier than both SFL schemes and AFL without adaptation. This demonstrates that the proposed method effectively mitigates the staleness issue inherent in AFL, thereby accelerating convergence in absolute time. Consequently, the proposed AFL framework achieves the best time-to-accuracy performance among all compared schemes, highlighting its advantage for time-constrained edge learning scenarios. The quantitative comparison of the time required to reach different accuracy levels is summarized in Table II.

E. Case Study of Model Staleness in AFL

Fig. 10 and Fig. 11 show the accuracy and loss curves of the global models in different FL frameworks.

As shown in Fig. 10 and Fig. 11, during the early training stage, the AFL model without the adaptive aggregation scheme exhibits significant fluctuations. This leads to unstable performance, with accuracy consistently lagging behind that of the SFL model. In contrast, at the corresponding nodes, the AFL model equipped with the adaptive aggregation scheme shows smaller fluctuations (e.g., Epoch 9 in Fig. 10a, Epoch 7 in Fig. 10b, Epoch 4 in Fig. 10c, and Epoch 6 in Fig. 10d). This is because the algorithm mitigates the impact of highly stale models. Furthermore, in certain cases, the global model can improve performance without being impeded by stale models (e.g., Epoch 6 in Fig. 10a, Epoch 10 in Fig. 10b, Epoch 9 in Fig. 10c, and Epoch 8 in Fig. 10d).

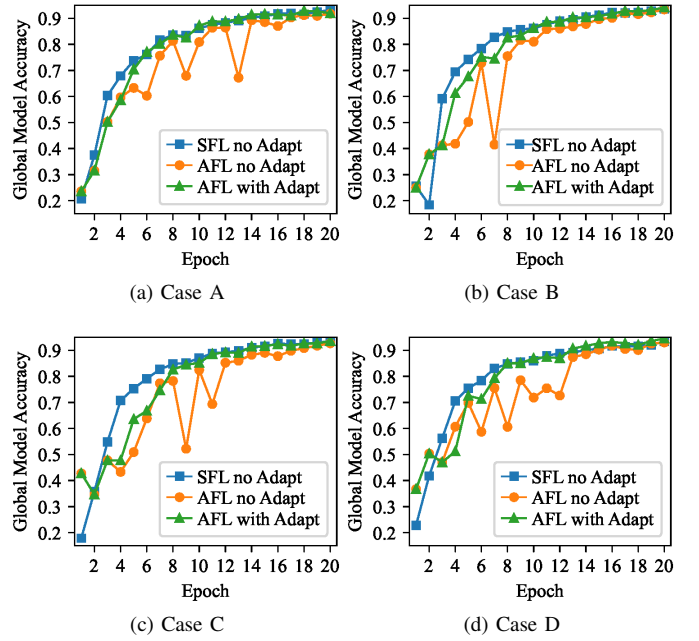


Fig. 10. Comparisons of accuracy curves in different FL frameworks. Our proposed self-adaptive aggregation scheme can prevent the dramatic fluctuations caused by model staleness.

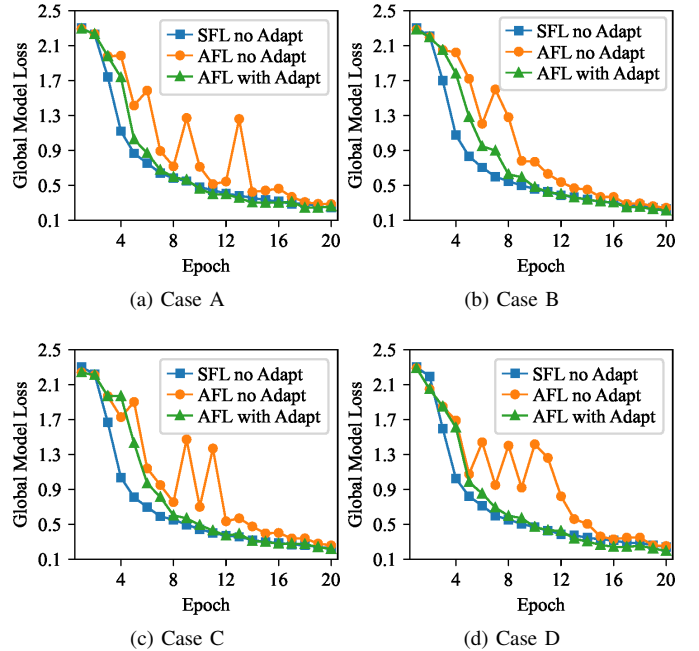


Fig. 11. Comparisons of loss curves in different FL frameworks. Our proposed self-adaptive aggregation scheme can prevent the dramatic fluctuations caused by model staleness.

These simulation results conclusively demonstrate that the proposed self-adaptive aggregation scheme effectively reduces the negative influence of model staleness in heterogeneous federated learning, accelerates the convergence rate of the global model, and enhances its final performance after convergence.

F. Case Study of Model Performance in FL

Fig. 12 shows the results of target detection completed by the models trained in different FL frameworks after the first

VI. CONCLUSION

In this paper, we present an AirComp-assisted AFL framework for UAV swarm networks that addresses the challenges of improving model-aggregation efficiency and mitigating model staleness. Our framework leverages UAVs as edge computing nodes to collect data and train local models, while swarm heads act as central PSs to aggregate local models. By integrating an energy-efficient AirComp transmission scheme, we enable simultaneous superposition of local model signals, optimizing communication time and bandwidth usage and formulate a data volume maximization problem under signal distortion and power constraints.

To solve the formulated problem, we decompose it into UAV selection and beamforming design subproblems, which are addressed using a branch-and-bound algorithm and our proposed alternating optimization algorithm, respectively. This approach balances computational complexity and performance, ensuring practical applicability in UAV swarm scenarios. Additionally, we propose a self-adaptive aggregation scheme that mitigates model staleness by enabling UAVs to upload only those layers of local models with high cosine similarity to the latest global model, significantly accelerating convergence.

Through simulations on practical deep learning tasks, we evaluate the proposed framework against conventional SFL systems with and without AirComp. The results demonstrate superior performance in model accuracy, loss, energy efficiency, and training data throughput, highlighting the effectiveness of our AirComp-assisted design and adaptive aggregation strategy. This work advances the integration of UAV swarms and federated learning, offering a robust solution for distributed edge computing with efficient resource utilization and convergence efficiency.

APPENDIX

PROOF OF THE CONVEXITY

To prove the convexity of subproblem \mathcal{P}_{2-1} , its optimization target and constraints should both be convex. Assume the complex vector \mathbf{b}_j consists of the real part $\mathbf{x} \in \mathbb{R}^A$ and the imaginary part $\mathbf{y} \in \mathbb{R}^A$:

$$\mathbf{b}_j = \mathbf{x} + i\mathbf{y}, \mathbf{x}, \mathbf{y} \in \mathbb{R}^A. \quad (51)$$

In this way, the objective function $\|\mathbf{b}_j\|^2$ can be expressed as:

$$\|\mathbf{b}_j\|^2 = \mathbf{b}_j^H \mathbf{b}_j = \mathbf{x}^T \mathbf{x} + \mathbf{y}^T \mathbf{y}. \quad (52)$$

Hence, the gradient of $\|\mathbf{b}_j\|^2$ with respect to \mathbf{x} and \mathbf{y} is

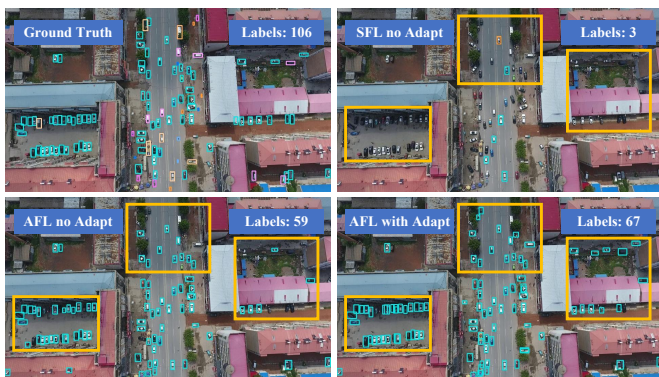
$$\nabla_{\mathbf{x}} \|\mathbf{b}_j\|^2 = 2\mathbf{x}, \nabla_{\mathbf{y}} \|\mathbf{b}_j\|^2 = 2\mathbf{y}. \quad (53)$$

The Hessian matrix is the matrix of second-order derivatives of the objective function for the decomposed real variables \mathbf{x} and \mathbf{y} . The second derivatives with respect to \mathbf{x} and \mathbf{y} are:

$$\frac{\partial^2}{\partial \mathbf{x} \partial \mathbf{x}^T} \|\mathbf{b}_j\|^2 = 2\mathbf{I}_A, \frac{\partial^2}{\partial \mathbf{y} \partial \mathbf{y}^T} \|\mathbf{b}_j\|^2 = 2\mathbf{I}_A \quad (54)$$

where \mathbf{I}_A is the $A \times A$ identity matrix. The cross-derivatives between \mathbf{x} and \mathbf{y} are given by:

$$\frac{\partial^2}{\partial \mathbf{x} \partial \mathbf{y}^T} \|\mathbf{b}_j\|^2 = \mathbf{0}, \frac{\partial^2}{\partial \mathbf{y} \partial \mathbf{x}^T} \|\mathbf{b}_j\|^2 = \mathbf{0}. \quad (55)$$



(a) Case A



(b) Case B

Fig. 12. Target detection performed by models in different FL frameworks. The thin rectangles in various colours indicate the targets detected by deep learning models and the yellow bold rectangles indicate regions with significantly different results. The numbers in the upper-right corner indicate the number of correctly detected targets.

hour of simulation time. The images from the VisDrone2019 dataset are captured by real UAVs and thus reflect the actual difficulties encountered in real-world scenarios. Thin rectangles in various colours indicate different kinds of detected targets, while the yellow bold rectangles highlight significant differences among the results. To facilitate direct evaluation, the numbers in the upper right corner of each figure indicate the number of correctly detected targets.

In both cases in Fig. 12, there are obvious omissions and false detections in the results produced by SFL. In contrast, the model trained in AFL without the self-adaptive aggregation scheme reduces some of these errors but still has room for improvement in completeness and accuracy. However, the model trained in AFL with the self-adaptive aggregation scheme achieves results much closer to the ground truth. This indicates that incorporating the self-adaptive aggregation scheme in AFL effectively enhances the model, making the detection results more reliable and closer to real-world conditions. Thus, this AFL framework can be effectively applied to practical object detection tasks in real-world scenarios.

Thus, the full Hessian matrix $\mathbf{H} \in \mathbb{R}^{2A \times 2A}$ of $\|\mathbf{b}_j\|^2$ is

$$\mathbf{H} = \begin{bmatrix} 2\mathbf{I}_A & \mathbf{0} \\ \mathbf{0} & 2\mathbf{I}_A \end{bmatrix}. \quad (56)$$

Since the Hessian matrix is symmetric and positive definite, the objective function $\|\mathbf{b}_j\|^2$ is a strongly convex function, which guarantees the unique global minimum.

In terms of the constraints, since

$$e^{i\theta_i} = \cos\theta_i + i\sin\theta_i \quad (57)$$

and assuming that

$$\mathbf{b}_j^H \mathbf{h}_{i,j} = a + ib, \text{ where } a = \text{Re}(\mathbf{b}_j^H \mathbf{h}_{i,j}), b = \text{Im}(\mathbf{b}_j^H \mathbf{h}_{i,j}), \quad (58)$$

the expression in the constraint can be transformed as

$$\begin{aligned} & \mathbf{b}_j^H \mathbf{h}_{i,j} \cdot e^{i\theta_i} \\ &= (a + ib)(\cos\theta_i + i\sin\theta_i) \\ &= (a\cos\theta_i - b\sin\theta_i) + i(a\sin\theta_i + b\cos\theta_i) \end{aligned} \quad (59)$$

and its real part is

$$\text{Re}(\mathbf{b}_j^H \mathbf{h}_{i,j} \cdot e^{i\theta_i}) = a\cos\theta_i - b\sin\theta_i. \quad (60)$$

In this way, the constraint is transformed into

$$a\cos\theta_i - b\sin\theta_i \geq \sqrt{\gamma_i}, \quad \forall i, j. \quad (61)$$

Since the inner product $\mathbf{b}_j^H \mathbf{h}_{i,j} = \sum_{k=1}^A b_{j,k}^* h_{i,j,k}$ is linear in the real and imaginary parts of \mathbf{b}_j , it follows that

$$a = \text{Re}(\mathbf{b}_j^H \mathbf{h}_{i,j}) = \mathbf{b}_j^{\text{Re}} \cdot \mathbf{h}_{i,j}^{\text{Re}} + \mathbf{b}_j^{\text{Im}} \cdot \mathbf{h}_{i,j}^{\text{Im}} \quad (62)$$

and

$$b = \text{Im}(\mathbf{b}_j^H \mathbf{h}_{i,j}) = \mathbf{b}_j^{\text{Re}} \cdot \mathbf{h}_{i,j}^{\text{Im}} - \mathbf{b}_j^{\text{Im}} \cdot \mathbf{h}_{i,j}^{\text{Re}} \quad (63)$$

where $\mathbf{b}_j = \mathbf{b}_j^{\text{Re}} + i\mathbf{b}_j^{\text{Im}}$ and $\mathbf{h}_{i,j} = \mathbf{h}_{i,j}^{\text{Re}} + i\mathbf{h}_{i,j}^{\text{Im}}$. Hence,

$$\begin{aligned} & \text{Re}(\mathbf{b}_j^H \mathbf{h}_{i,j} \cdot e^{i\theta_i}) \\ &= (\mathbf{b}_j^{\text{Re}} \cdot \mathbf{h}_{i,j}^{\text{Re}} + \mathbf{b}_j^{\text{Im}} \cdot \mathbf{h}_{i,j}^{\text{Im}})\cos\theta_i - (\mathbf{b}_j^{\text{Re}} \cdot \mathbf{h}_{i,j}^{\text{Im}} - \mathbf{b}_j^{\text{Im}} \cdot \mathbf{h}_{i,j}^{\text{Re}})\sin\theta_i, \end{aligned} \quad (64)$$

indicating that $\text{Re}(\mathbf{b}_j^H \mathbf{h}_{i,j} \cdot e^{i\theta_i})$ is an affine function with respect to \mathbf{b}_j^{Re} and \mathbf{b}_j^{Im} . Therefore, the constraint turns out to be a linear inequality that defines a half-space in the real vector space \mathbb{R}^{2A} . Since a half-space is naturally convex, each constraint defines $\text{Re}(\mathbf{b}_j^H \mathbf{h}_{i,j} \cdot e^{i\theta_i})$ as a convex feasible set and their intersection also defines a convex feasible set.

In conclusion, since the objective function and the constraints are both convex, the optimization problem is convex.

REFERENCES

[1] Y. Zhao *et al.*, “Joint content caching, service placement, and task offloading in UAV-enabled mobile edge computing networks,” *IEEE J. Sel. Areas Commun.*, vol. 43, no. 1, pp. 51–63, Jan. 2025.

[2] J. Tang *et al.*, “Cooperative ISAC-empowered low-altitude economy,” *IEEE Trans. on Wireless Commun.*, vol. 24, no. 5, pp. 3837–3853, May 2025.

[3] R. Huang *et al.*, “Dynamic task offloading for multi-UAVs in vehicular edge computing with delay guarantees: A consensus ADMM-based optimization,” *IEEE Trans. Mob. Comput.*, vol. 23, no. 12, pp. 13 696–13 712, Dec. 2024.

[4] S. Shahriar Ahmed *et al.*, “The state of urban air mobility research: An assessment of challenges and opportunities,” *IEEE Trans. Intell. Transp. Syst.*, vol. 26, no. 2, pp. 1375–1394, Feb. 2025.

[5] J. Li *et al.*, “UAV-assisted microservice mobile edge computing architecture: Addressing post-disaster emergency medical rescue,” *IEEE Trans. Comput.*, vol. 74, no. 8, pp. 2635–2648, Aug. 2025.

[6] S. Park, C. Park, and J. Kim, “Learning-based cooperative mobility control for autonomous drone-delivery,” *IEEE Trans. Veh. Technol.*, vol. 73, no. 4, pp. 4870–4885, Apr. 2024.

[7] M. Song *et al.*, “Trustworthy intelligent networks for low-altitude economy,” *IEEE Commun. Mag.*, vol. 63, no. 7, pp. 72–79, Jul. 2025.

[8] X. Xu *et al.*, “Joint UAV deployment and resource allocation: A personalized federated deep reinforcement learning approach,” *IEEE Trans. Veh. Technol.*, vol. 73, no. 3, pp. 4005–4018, Mar. 2024.

[9] W. He *et al.*, “Three-stage stackelberg game enabled clustered federated learning in heterogeneous UAV swarms,” *IEEE Trans. Veh. Technol.*, vol. 72, no. 7, pp. 9366–9380, Jul. 2023.

[10] M. Zhao *et al.*, “IC2S-swarm: When digital twin meets collaborative ISR,” *IEEE Commun. Mag.*, vol. 63, no. 4, pp. 221–227, Apr. 2025.

[11] W. Y. B. Lim *et al.*, “Federated learning in mobile edge networks: A comprehensive survey,” *IEEE Commun. Surv. Tutorials*, vol. 22, no. 3, pp. 2031–2063, Apr. 2020.

[12] A. Perez-Portero *et al.*, “Airborne GNSS-R: A key enabling technology for environmental monitoring,” *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 14, pp. 6652–6661, Apr. 2021.

[13] Y. Liu *et al.*, “Vertical federated learning: Concepts, advances, and challenges,” *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 7, pp. 3615–3634, Jul. 2024.

[14] Y. Peng *et al.*, “How to tame mobility in federated learning over mobile networks?” *IEEE Trans. on Wireless Commun.*, vol. 22, no. 12, pp. 9640–9657, Dec. 2023.

[15] X. Liu, Y. Deng, and T. Mahmoodi, “Wireless distributed learning: A new hybrid split and federated learning approach,” *IEEE Trans. on Wireless Commun.*, vol. 22, no. 4, pp. 2650–2665, Apr. 2023.

[16] P. Huang, D. Li, and Z. Yan, “Wireless federated learning with asynchronous and quantized updates,” *IEEE Commun. Lett.*, vol. 27, no. 9, pp. 2393–2397, Sept. 2023.

[17] C. Xu *et al.*, “An efficient and reliable asynchronous federated learning scheme for smart public transportation,” *IEEE Trans. Veh. Technol.*, vol. 72, no. 5, pp. 6584–6598, May 2023.

[18] Z. Yan and D. Li, “Convergence time optimization for decentralized federated learning with LEO satellites via number control,” *IEEE Trans. Veh. Technol.*, vol. 73, no. 3, pp. 4517–4522, Mar. 2024.

[19] D. Qiao *et al.*, “ASMAFL: Adaptive staleness-aware momentum asynchronous federated learning in edge computing,” *IEEE Trans. Mob. Comput.*, vol. 24, no. 4, pp. 3390–3406, Apr. 2025.

[20] Z. Lin *et al.*, “FedSN: A federated learning framework over heterogeneous LEO satellite networks,” *IEEE Trans. Mob. Comput.*, vol. 24, no. 3, pp. 1293–1307, Mar. 2025.

[21] Q. Wu *et al.*, “HiFlash: Communication-efficient hierarchical federated learning with adaptive staleness control and heterogeneity-aware client-edge association,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 34, no. 5, pp. 1560–1579, May 2023.

[22] Y. Wan *et al.*, “Low-light and infrared multimodal remote sensing in nighttime rescue mission: A review of anomaly detection methods,” *IEEE Trans. Geosci. Remote Sens.*, vol. 63, pp. 1–18, May 2025.

[23] S. Zhang and J. Liu, “Analysis and optimization of multiple unmanned aerial vehicle-assisted communications in post-disaster areas,” *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 12 049–12 060, Dec. 2018.

[24] Z. Lin *et al.*, “Deep convolutional highway unit network for SAR target classification with limited labeled training data,” *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 7, pp. 1091–1095, Jul. 2017.

[25] Z. Zhou *et al.*, “Towards efficient and stable k-asynchronous federated learning with unbounded stale gradients on Non-IID data,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 12, pp. 3291–3305, Dec. 2022.

[26] W. Liu *et al.*, “Over-the-air computation systems: Optimization, analysis and scaling laws,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 8, pp. 5488–5502, Aug. 2020.

[27] G. Zhu *et al.*, “One-bit over-the-air aggregation for communication-efficient federated edge learning: Design and convergence analysis,” *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 2120–2135, Mar. 2021.

[28] X. Cao *et al.*, “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.

[29] K. Yang *et al.*, “Federated learning via over-the-air computation,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 2022–2035, Mar. 2020.

[30] A. Mohammad 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.

- [31] A. Mohammad and D. Gündüz, "Federated learning over wireless fading channels," *IEEE Trans. on Wireless Commun.*, vol. 19, no. 5, pp. 3546–3557, Jan. 2020.
- [32] S. Sun *et al.*, "Staleness-controlled asynchronous federated learning: Accuracy and efficiency tradeoff," *IEEE Trans. Mob. Comput.*, vol. 23, no. 12, pp. 12 621–12 634, Dec. 2024.
- [33] H. Zhang *et al.*, "Federated multi-task learning with non-stationary and heterogeneous data in wireless networks," *IEEE Trans. Wireless Commun.*, vol. 23, no. 4, pp. 2653–2667, Apr. 2024.
- [34] Y. Xiao *et al.*, "Time-sensitive learning for heterogeneous federated edge intelligence," *IEEE Trans. Mob. Comput.*, vol. 23, no. 2, pp. 1382–1400, Feb. 2024.
- [35] F. Liang *et al.*, "Semi-synchronous federated learning protocol with dynamic aggregation in internet of vehicles," *IEEE Trans. Veh. Technol.*, vol. 71, no. 5, pp. 4677–4691, May 2022.
- [36] S. Fang *et al.*, "Zero forcing assisted single layer beamforming for spatial modulation MIMO systems," *IEEE Trans. Veh. Technol.*, vol. 71, no. 4, pp. 4116–4128, Apr. 2022.
- [37] J. Wang and G. Joshi, "Cooperative SGD: A unified framework for the design and analysis of local-update SGD algorithms," *J. Mach. Learn. Res.*, vol. 22, no. 1, pp. 9709–9758, Jan. 2021.
- [38] R. Talak, S. Karaman, and E. Modiano, "Improving age of information in wireless networks with perfect channel state information," *IEEE/ACM Trans. Networking*, vol. 28, no. 4, pp. 1765–1778, Aug. 2020.
- [39] R. Wang, X. Yuan, and J. Wu, "Degrees of freedom of a MIMO multipair two-way relay channel with delayed channel state information," *IEEE Signal Process Lett.*, vol. 25, no. 2, pp. 243–247, Feb. 2018.
- [40] S. Gong *et al.*, "Energy efficient transmission in multi-user MIMO relay channels with perfect and imperfect channel state information," *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 3885–3898, Jun. 2017.
- [41] C. Luo *et al.*, "Channel state information prediction for 5G wireless communications: A deep learning approach," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 227–236, Jan. 2020.
- [42] Z. Xiao *et al.*, "Nonparametric regression for MU-MIMO channel prediction: From KNN to local linear regression," *IEEE Trans. Wireless Commun.*, vol. 23, no. 4, pp. 2784–2795, Apr. 2024.
- [43] C. Lu and Y. Liu, "An efficient global algorithm for single-group multicast beamforming," *IEEE Trans. Signal Process.*, vol. 65, no. 14, pp. 3761–3774, Jul. 2017.
- [44] Y. Huang *et al.*, "Asynchronous federated learning via over-the-air computation in LEO satellite networks," *IEEE Trans. on Wireless Commun.*, vol. 23, no. 12, pp. 19 885–19 901, Dec. 2024.
- [45] Z. Zhao *et al.*, "Federated learning with Non-IID data in wireless networks," *IEEE Trans. Wireless Commun.*, vol. 21, no. 3, pp. 1927–1942, Mar. 2022.
- [46] P. Chen and X. Ma, "Optimal strategy for designing a multitask learning-based hybrid model to predict wheat leaf nitrogen content," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, pp. 1–5, Sept. 2023.
- [47] J. Kaplan *et al.*, "Scaling laws for neural language models," *arXiv preprint arXiv:2001.08361*, Jan. 2020.
- [48] Y. Lecun *et al.*, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
- [49] P. Zhu *et al.*, "Detection and tracking meet drones challenge," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 7380–7399, Nov. 2021.



include AI-driven networks, UAV swarm network management, and human-swarm interaction.



Lu Zhang received the B.S. degree in Electronic Information Engineering from Nanjing University of Posts and Telecommunications, China, in 2023. She is currently pursuing her master's degree in Communication Engineering at Beijing University of Posts and Telecommunications (BUPT), China. Her research interests include UAV swarm networks and AI-driven networks.



Mugen Peng (Fellow, IEEE) received the Ph.D. degree in communication and information systems from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2005. Afterwards, he joined BUPT, where he has been a Full Professor with the School of Information and Communication Engineering since 2012. In 2014, he was an Academic Visiting Fellow with Princeton University, USA. He leads a Research Group focusing on wireless transmission and networking technologies with the State Key Laboratory of Networking and Switching Technology, BUPT. He was a recipient of the 2018 Heinrich Hertz Prize Paper Award, the 2014 IEEE ComSoc AP Outstanding Young Researcher Award, and the Best Paper Award in the JCN 2016. He has been on the Editorial/Associate Editorial Board of the IEEE Communications Magazine, the Internet of Things Journal, and IEEE ACCESS.



Yansong Huang is currently a Ph.D. student in Electronic and Electrical Engineering at University College London, London, United Kingdom. He received the M.S. degree in Information and Communication Engineering from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2025. Before that, he received the B.S. degree in Telecommunications Engineering with Management from BUPT, Beijing, China, in 2022. His current research interests include federated learning and fluid antenna systems.