Asynchronous Federated Learning via Over-the-Air Computation in LEO Satellite Networks

Yansong Huang, Xuan Li, Moke Zhao, Haiyan Li, Mugen Peng
Beijing University of Posts and Telecommunications, China

***Abstract*—Owing to its ability to offer collaborative data utilization while ensuring data privacy, federated learning (FL) provides a promising paradigm to enable cooperative intelligent tasks across multiple low-earth orbit (LEO) satellites, such as carbon estimation, traffic surveillance, and forest fire detection. Although the advantages of pushing intelligence to satellites are multi-fold, limited communication channels along with the rigid global model aggregation conditions result in dramatic convergence delays. In order to reduce the convergence time, we propose an asynchronous FL framework in LEO satellite networks by exploiting multiple high-altitude platforms for model aggregation, where the advanced over-the-air computation (AirComp) transmission scheme is utilized for the sake of further reducing energy consumption. Considering the practical constraint of AirComp signal distortion, the objective function of optimizing FL performance is carefully formulated and solved by the proposed quantity-quality jointed linkage search algorithm. Simulation results demonstrate that our proposed asynchronous FL framework outperforms the conventional synchronous FL framework by a decline of 30.07% in convergence time at most. It also provides an average increase of 110% and 580%, respectively, in terms of throughput and energy efficiency in all scenarios considered. Overall, our study presents a beneficial asynchronous FL framework and a fast aggregation scheduling algorithm in LEO satellite networks, accelerating the convergence of the global model with reduced energy expenditure.**

***Index Terms*—Low-earth orbit satellite networks, asynchronous federated learning, over-the-air computation.**

I. Introduction

A. Background

THE rapid expansion of low earth orbit (LEO) satellite networks, such as Starlink, is poised to significantly enhance satellite communications with reduced latency and increased bandwidth [1]–[4]. This global communication capability and abundant data resources present new opportunities to integrate computing resources and data within LEO satellite networks for goal-oriented tasks based on deep learning technology [5], such as carbon estimation [6], transportation surveillance [7], and forest fire detection [8]. However, implementing these tasks in a centralized deep learning pattern remains challenging due to the voluminous size of raw data collected by satellites and the potential threat of privacy breaches.

In this context, federated learning (FL) emerges as a promising distributed machine learning approach that allows for model training on local devices while preserving privacy [9]–[11]. In the scenario of a LEO satellite network, FL conserves communication bandwidth by enabling edge satellites to transmit the learned machine learning model parameters, rather than transmitting the massive raw data. Additionally, FL makes it difficult for malicious hackers or other FL participants to reverse-engineer the content of training data from these updated parameters, thus safeguarding the raw data from potential leaks [12]. Nevertheless, substantial challenges remain in the application of FL to LEO satellite networks [13], [14]. The primary challenge lies in the convergence rate, which is influenced by both the number of learning iterations and the time required for each iteration [15].

Reducing the number of learning iterations is challenging as a large deep learning model with millions of parameters typically requires several hundred iterations to converge [16]. Besides, because of the restricted computing capabilities of LEO satellites and the large amount of training data they collect, the time of training a deep learning model at a satellite is much longer than that on a terrestrial client. Moreover, the large speed of satellites leads to the short visible window to the parameter servers (PS), which aggregates the updated model parameters from satellites and is typically deployed at a ground station, also affecting the convergence rate [17]. The satellite overpass time is usually short and periodic due to the differences between the travel trajectories of satellites and the PS, as well as the variation in satellite orbiting speeds and the earth rotation speed. In this case, the intermittent and sporadic connections prolong the time for parameter servers to collect local models for satellites. Consequently, the traditional FL process takes several days, at the very least, to converge its deep learning model.

B. Motivation

In some emergency response tasks, remote sensing images are used to analyze the situations of the accidents. When deep learning is introduced into such tasks to relieve the workload from human beings, it is expected to support decision-making in a shorter time because of the emergency, even at a little cost of accuracy. However, when a deep learning model is applied to a new scenario, it usually needs to be finetuned to adapt itself to this scenario by training on the dataset made up of the data of the ongoing accidents. In this case, the convergency time of the global model in FL is expected to be shorter.

To address the problem of low convergence rate, asynchronous FL has been proposed to relax the aggregation condition [18]. In the asynchronous FL scheme, the PS aggregates parameters from accessible satellites at fixed intervals, rather than waiting for all satellites to transmit their parameters [7]. By eliminating inefficient waiting time, the iteration time is significantly reduced. However, this reduction also leads to a

decrease in 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 [19]. Besides, the staleness issue in asynchronous FL caused by delayed global model updating in satellites may lead to slow improvement in model performance and even weight divergence [20]. To address the drawbacks of decreased training data and staleness issue, the data volume size coefficient and staleness coefficient are applied to local models in the aggregation procedure. To apply asynchronous FL to LEO satellite networks, we propose a feasible framework which deploys PS at several high-altitude platforms (HAPs). They are semi-static aircraft located in the stratosphere, approximately 17-22 kilometers (km) above the surface of the earth [21]. They provide more stable and extensive communication coverage than ground stations, enhancing signal reliability and reach. Leveraging a ring topology for HAPs and star topology for HAPs and satellites, this asynchronous FL framework accelerates model convergence by decreasing the idle time of satellites, resulting in more efficient FL tasks.

In order to further speed up the model aggregation, the advanced over-the-air computation (AirComp) transmission scheme may be conceived in the FL framework, by taking advantage of simultaneously processing communication and computation [22]. It explores signal superposition of weighted model parameters in the wireless multiple-access channel and thus supports less energy consumption by integrating the receiving of multiple local models [23]. However, despite the fact that the AirComp technique utilizes the signal superposition to avoid the interchannel interference, the signal is still affected by other factors, resulting in distortion [24], which can hardly be detected and corrected by the receiver for it just attains the overlap of signals from multiple transmitters and the distortion are mixed as well. Hence the beamforming vector of the receiver should be carefully designated to eliminate the channel interference in respect to each transmitter. Sometimes satellites with poor channel conditions must be removed and excluded from global aggregation in a given epoch. This contradicts the aim of the asynchronous FL framework, which seeks to include as many satellites as possible in each epoch to utilize more training data.

C. Related Work

Existing research on FL combined with AirComp primarily focuses on terrestrial networks. These studies, such as those by Mohammadi Amiri and Gündüz [25], [26], and by Zhu et al. [27], [28], have demonstrated the effectiveness of these technologies in enhancing communication efficiency and reducing latency in edge learning scenarios. However, their direct application to LEO satellite networks is not straightforward. LEO satellites 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. The intricate dynamics of satellite movement introduce additional complexities such as variable Doppler effects, which are not typically encountered in terrestrial settings.

Some research has started to adapt FL techniques to satellite scenarios, acknowledging the specific limitations such as constrained computational power on satellites. These studies have employed various methods to reduce the convergence time of global models, which seems somewhat restricted by the synchronous FL mechanisms typically used. Elmahallawy et al. [29] introduced an FL approach that integrates non-orthogonal multiple access (NOMA) with LEO satellites and HAPs to enhance model transmission and aggregation efficiency.

To further reduce the convergence time of global models, some studies have incorporated asynchronous FL techniques in LEO scenarios, demonstrating promising results. However, these approaches often overlook the potential for greater utilization of communication resources, which could further enhance performance and efficiency. For instance, Elmahallawy and Luo [21] presented an asynchronous FL framework for LEO constellations that effectively addressed issues of model staleness and communication bottlenecks but did not fully explore the use of communication resources.

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

D. Contributions and Organization

We design an asynchronous FL framework with AirComp technique in LEO satellite networks, while jointly solving linkage scheme and beamforming vector. To be more specific,

- we propose an efficient asynchronous FL framework, where the satellites act as edge computation nodes and HAPs function as central parameter servers. It allows part of satellites to participate in aggregation in each epoch instead of all of them, thereby sharply reducing the time required for global model convergence. The beneficial cooperation of satellites and HAPs constitutes the basis of a structurally efficient FL framework.
- We design an energy-efficient AirComp transmission scheme through combining the updates from multiple satellites in a single communication round, and carefully formulate its data volume maximization problem by taking into account the practical constraints of signal distortion, power limitation, etc.
- We find the optimal solution of this joint linkage search and beamforming vector problem by a high-complexity exhaustive search first which may have an overwhelming complexity even for a modest-scale system. To reduce the computational complexity, we propose a heuristic quantity-quality joint linkage search algorithm combining depth-first search (DFS) and breadth-first search (BFS)

algorithm with subtree pruning. Explicitly, the DFS algorithm devotes to generating the best linkage scheme in each epoch, while the BFS algorithm focuses on solving the beamforming vectors for HAPs.

- We evaluate the proposed asynchronous FL framework by implementing a practical deep learning task and compare our design- in terms of its achievable model accuracy, global loss, energy efficiency, and training data throughput to the conventional synchronous FL system with and without AirComp transmission scheme.

The rest of our paper is organized as follows: Section II demonstrates the asynchronous FL framework working in the LEO satellite networks with AirComp, focusing on the model aggregation flow. Section III states the optimization problem to be solved in this framework, and proposes our optimal and suboptimal solution based on DFS and BFS algorithm with subtree pruning. Section IV presents the simulation results of the asynchronous FL framework in various indicators. Section V analyzes the results and concludes our work.

II. System Model

In this section, we introduce an asynchronous FL framework for LEO satellite networks including LEO satellite networks as edge computation nodes and HAPs as central parameter servers. Additionally, we utilize AirComp in model aggregation for the sake of further reducing energy consumption. The constraint in communication quality brought out by AirComp is modeled and illustrated in detail.

A. Asynchronous FL Framework for LEO Satellite Networks

Consider an LEO satellite network consisting of single-antenna satellites M whose orbits do not intersect with each other, along with HAPs N equipped with L antennas. The satellites follow the same orbits as those in Starlink, while the HAPs are evenly distributed along the equator and remain stationary relative to the earth. In this way, satellites form a star topology with a HAP working as a central hub to transmit local models and receive global models.

In this scenario, direct communication between satellites is not feasible due to their high-speed movement and unpredictable Doppler shifts, which result in large and variable frequency offsets that are difficult to compensate given the limited power and computational resources of satellites [30]. But the Doppler shift between HAPs and satellites can be more effectively managed due to the advanced technologies equipped on HAPs and their relatively stable positions compared to moving satellites [31].

Each HAP is connected to its two adjacent neighbors, forming a ring topology, as illustrated in Fig. 1. Among all the HAPs, two of them are randomly assigned as the start HAP and end HAP in each epoch. The updated local models transmitted from satellites are aggregated from the start HAP to the end HAP, forming the new global model. In this manner, regardless of the HAP where satellites are connected, their locally uploaded models can be aggregated into the global model, which is then relayed in reverse order to all connected satellites and works as their new model in the subsequent

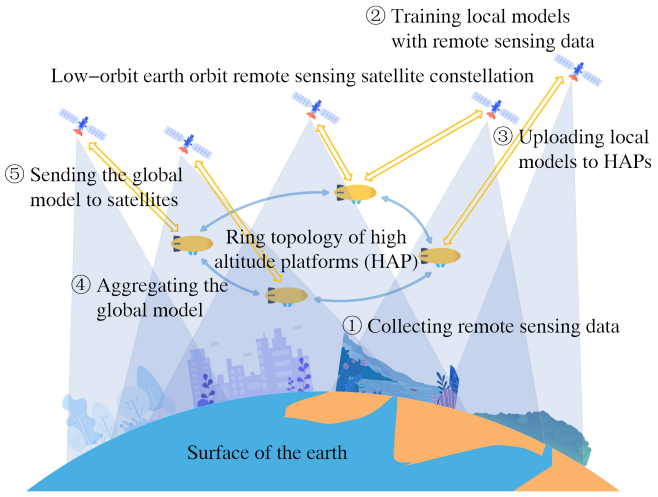


Fig. 1. Illustraition of the asynchronous FL framework.

epoch. Satellites that fail to establish a connection with any HAP in a particular epoch continue training based on the current model until they link with the HAPs.

As components of the framework, satellites and HAPs work together to accomplish the deep learning task in epoch t by minimizing the loss function of the global model:

$$\underset{\boldsymbol{z}^{[t]}\in\mathbb{R}^w}{\text{minimize}}\ \mathcal{J}(\boldsymbol{z}^{[t]}) = (1-\rho)\mathcal{J}(\boldsymbol{z}^{[t-1]}) + \frac{1}{\psi}\sum_{m\in M^{[t]}}\phi_m\mathcal{J}_m(\boldsymbol{z}_m^{[t]}), \tag{1}$$

where $\boldsymbol{z}^{[t]}$ is the global model aggregated in epoch t by global model $\boldsymbol{z}^{[t-1]}$ of last epoch $t-1$ and local model $\boldsymbol{z}_m^{[t]}$ from satellite $m \in M^{[t]}$. $M^{[t]}$ indicates the satellites participate in the aggregation of epoch t. ϕ_m and ψ are pre- and post-processing scalar at transmitter and receiver respectively, evaluating the weights of each model. ρ is the mix of pre- and post-processing scalars simply given as:

$$\rho = \frac{1}{\psi}\sum_{m\in M^{[t]}}\phi_m. \tag{2}$$

The preprocessing scalar ϕ_m is defined as

$$\phi_m = |d_m|\frac{\boldsymbol{\beta}_m}{t+1}, \tag{3}$$

where local dataset $d_m \subseteq \mathcal{D}$ indicates the data utilized by each satellite internally and global dataset $\mathcal{D} = \{(\boldsymbol{x}_m, y_m) : m \in M\}$ includes training data utilized by all satellites in epoch t. The local model $\boldsymbol{z}_m^{[t]}$ is trained with data d_m as

$$\boldsymbol{z}_m^{[t+1]} = \boldsymbol{z}^{[t]} - \varpi\nabla\mathcal{J}_m(\boldsymbol{z}^{[t]}), \tag{4}$$

where ϖ indicates the learning rate and $\nabla\mathcal{J}_m(\cdot)$ is the gradient of the loss function.

Considering the heterogeneity between various satellites and thus distinction in training data and the chance to update global models, its effects on the performance of local models should be assessed in the aspect of data volume and model staleness. To evaluate these two factors, satellite $m \in M^{[t]}$ applies pre-processing scalar $|d_m|$ and $\boldsymbol{\beta}_m/(t+1)$ to its local model $\boldsymbol{z}_m^{[t]}$. The first scalar $|d_m|$ indicates data volume and the second one $\boldsymbol{\beta}_m/(t+1)$ is the ratio between the epoch of the

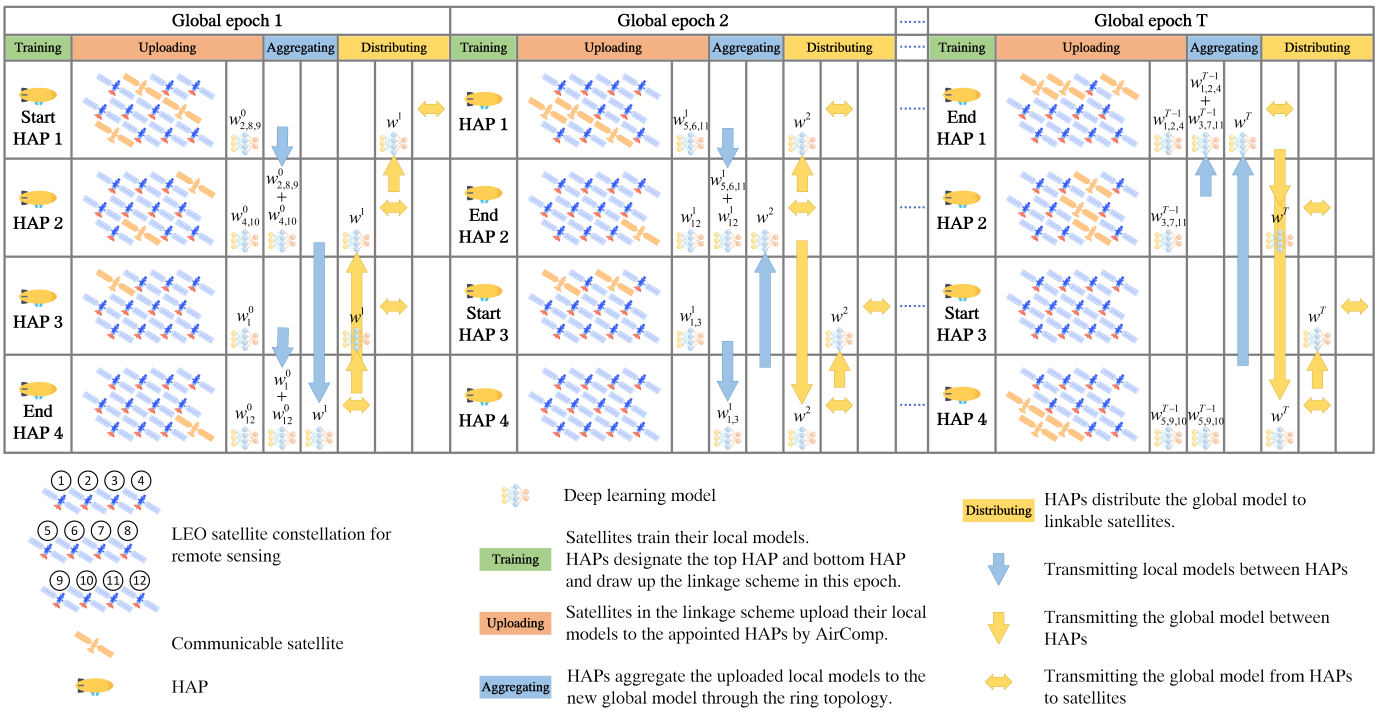


Fig. 2. Sequential diagram of the asynchronous FL framework.

global model $\boldsymbol{\beta}_m$ that satellite m bases on to train and the epoch of the aggregating global model $t+1$. Therefore, the larger the data volume a satellite applies to training its model and the fresher global model it utilizes as the initial model, its local model gets better performance and thus deserves higher weight than others in the aggregation procedure. After accumulating uploaded local models, post-processing scalar $1/(\sum_{m\in M^{[t]}}|d_m|)$ is applied to them to offset amplification from pre-processing scalar and then get new global models.

This framework, as demonstrated in Fig. 1, is divided into four stages: training, uploading, aggregating and distributing. In the training stage, all satellites collects data through their sensors and continuously train their local models. Meanwhile, the HAPs work out the linkage scheme that tells satellites whether they should upload their local models in this epoch and which HAP should they send models to. In the uploading stage, satellites transmit their models together through AirComp and therefore a HAP receives superposed models. The ring topology helps HAPs aggregate the global model from the start HAP to the end one in aggregating stage. Finally, the global model is shared to all HAPs from the end one to the start one and then sended to communicatable satellites in the distributing stage. The whole model propagation algorithm working in our proposed asynchronous FL framework is demonstrated in Algorithm 1 with detail.

Algorithm 1 Model Propagation Algorithm

Initialize Global model $\boldsymbol{z}^{[0]}$
for epoch $t=0,\cdots,T$ **do**
  Select start HAP n_{start} and end HAP n_{end} randomly
  n_{start} updates the global model $\boldsymbol{z}^{[t]}$ to all HAPs
  for each HAP $n\in N$ **do**
    Transmit $\boldsymbol{z}^{[t]}$ to communicatable satellites
  end for
  for each satellite $m\in M$ **do**
    Obtain d_m through sensors
    Training models $\boldsymbol{z}_m^{[t+1]}=\boldsymbol{z}^{[t]}-\varpi\nabla\mathcal{J}_m(\boldsymbol{z}^{[t]})$
  end for
  n_{end} generates linkage scheme by Algorithm 2
  Satellites transmit local models $\boldsymbol{z}_m^{[t+1]}$ to HAPs
  for each HAP $n\in N$ **do**
    Receive models $\boldsymbol{z}_n^{[t+1]}=\frac{1}{\psi}\sum\phi_m\mathcal{J}_m(\boldsymbol{z}_m^{[t+1]})$ from satellites $m\in M_n$
  end for
  n_{end} collects models from all HAPs and generates new global model $\boldsymbol{z}^{[t+1]}=(1-\rho)\boldsymbol{z}^{[t]}+\sum\boldsymbol{z}_n^{[t+1]}$
end for

B. Over-The-Air Computation for Model Aggregation

In a traditional FL framework, edge nodes should relay their updated local model to central parameter servers separately, therefore restricted by limited bandwidth and consuming more energy [32]. To take full advantage of communication resources, AirComp is introduced to integrate computation and communication by exploiting the signal superposition property of a multiple-access channel and calculating the linear function of distributed models from various satellites [33]. As mentioned in Section II-A, a middle-aggregated model contributed by satellites M_n at HAP n is simply given by

$$\boldsymbol{z}_n=\frac{1}{\psi}\sum_{m\in M_n}\phi_m\boldsymbol{z}_m. \tag{5}$$

Assume the signal vector for each local model $\boldsymbol{p}_m:=\boldsymbol{z}_m\in$

$\mathbb{C}^d$ to be normalized with unit variance, i.e., $\mathbb{E}(\boldsymbol{p}_m \boldsymbol{p}_m^{\mathsf{H}}) = \boldsymbol{I}$. Here the operation $\boldsymbol{p}_m^{\mathsf{H}}$ is to get the conjugate transpose matrix of the original one. At each time slot $j \in \{1, \cdots, d\}$, satellites $m \in M_n$ sends the signal $\boldsymbol{p}_m^{(j)} \in \mathbb{C}$ to HAP n through AirComp and the ideal received signal $p_n^{(j)}$ is denoted as

$$p_n^{(j)} = \sum_{m \in M_n} \kappa_m p_m^{(j)} \tag{6}$$

with $\kappa_m \in \mathbb{C}$ as transmitter scalar. To simplify the notation, the index of time slot j is omitted and the signal passing through the channel is presented as

$$\boldsymbol{y}_n = \sum_{m \in M_n} \boldsymbol{h}_{m,n} \kappa_m p_m + \boldsymbol{\zeta}, \tag{7}$$

where $\boldsymbol{h}_{m,n} \in \mathbb{C}^L$ is the channel vector between satellite m and HAP n and $\boldsymbol{\zeta} \sim \mathcal{CN}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$ is the noise vector.

The constraint of power at satellites is given as

$$\mathbb{E}(|\kappa_m p_m|^2) = |\kappa_m|^2 \leq P_{\text{sat}} \tag{8}$$

with $P_{\text{sat}} > 0$ as the maximum transmit power of satellites. The estimated signal received at HAP n with its beamforming vector $\boldsymbol{b}_n \in \mathbb{C}^L$ is presented as

$$\widehat{p}_n = \frac{1}{\sqrt{\eta_n}} \boldsymbol{b}_n^{\mathsf{H}} \boldsymbol{y}_n = \frac{1}{\sqrt{\eta_n}} \boldsymbol{b}_n^{\mathsf{H}} \sum_{m \in M_n} \boldsymbol{h}_{m,n} \kappa_m p_m + \frac{\boldsymbol{b}_n^{\mathsf{H}} \boldsymbol{\zeta}}{\sqrt{\eta_n}} \tag{9}$$

where η_n works as a normalizing factor.

The elements of beamforming vector $b_i \in \boldsymbol{b}$ all come from a designated codebook $\boldsymbol{W}$ designed as [34]

$$W_{a,p} = \frac{1}{2^{a-1}} e^{j \frac{2\pi p}{P}} \tag{10}$$

where $a = 1, \ldots, A$, $p = 1, \ldots, P$. The antenna power constraint for $|L|$ antennas at HAP n is given as

$$\left| b_l^{(n)} \right|^2 = b_l^{(n)} \overline{b_l^{(n)}} \leq \frac{P_{\text{HAP}}}{|L|}, \ \forall l, \tag{11}$$

where $b_l^{(n)} \in \boldsymbol{b}_n$ is the complex amplitude applied to the corresponding element $l \in L$ in the antenna array. Equation (11) refers that all antennas of a HAP have the same power limitation whose sum is P_{HAP} and it is highly related with the codewords $W_{a,p}$ in the codebook $\boldsymbol{W}$ as

$$P_{\text{HAP}} = L * \arg \sup_{W_{a,p} \in \boldsymbol{W}} \|W_{a,p}\|^2 = L. \tag{12}$$

The distortion between ideal signal p_n in (6) and estimated signal $\widehat{p}_n$ in (9), which qualifies the performance of AirComp, is measured by the mean-squared-error (MSE) defined as

$$\begin{aligned} \mathsf{MSE}(\widehat{p}_n, p_n) &= \mathbb{E}(|\widehat{p}_n - p_n|^2) \\ &= \sum_{m \in M_n} \left| \frac{\boldsymbol{b}_n^{\mathsf{H}} \boldsymbol{h}_{m,n} \kappa_m}{\sqrt{\eta_n}} - \phi_m \right|^2 + \sigma^2 \frac{\|\boldsymbol{b}_n\|^2}{\eta_n}. \end{aligned} \tag{13}$$

Given arbitrarily chosen receiver beamforming vector $\boldsymbol{b}_n$, the optimal transmitter scalar that minimizes the MSE is given by the following zero-forcing transmitter [35]:

$$\frac{\boldsymbol{b}_n^{\mathsf{H}} \boldsymbol{h}_{m,n} \kappa_m}{\sqrt{\eta_n}} - \phi_m = 0 \tag{14}$$

and the optimal transmitter scalar is given as

$$\kappa_m = \sqrt{\eta_n} \phi_m \frac{(\boldsymbol{b}_n^{\mathsf{H}} \boldsymbol{h}_{m,n})^{\mathsf{H}}}{\left\| \boldsymbol{b}_n^{\mathsf{H}} \boldsymbol{h}_{m,n} \right\|^2}. \tag{15}$$

Based on the transmission power constraint (6), it follows that

$$\left| \sqrt{\eta_n} \phi_m \frac{(\boldsymbol{b}_n^{\mathsf{H}} \boldsymbol{h}_{m,n})^{\mathsf{H}}}{\left\| \boldsymbol{b}_n^{\mathsf{H}} \boldsymbol{h}_{m,n} \right\|^2} \right|^2 \leq P_{\text{sat}}. \tag{16}$$

Given that $\boldsymbol{b}_n^{\mathsf{H}} \boldsymbol{h}_{m,n} \in \mathbb{C}$, it can be derived that

$$\left| (\boldsymbol{b}_n^{\mathsf{H}} \boldsymbol{h}_{m,n})^{\mathsf{H}} \right|^2 = \left\| (\boldsymbol{b}_n^{\mathsf{H}} \boldsymbol{h}_{m,n})^{\mathsf{H}} \right\|^2 = \left\| \boldsymbol{b}_n^{\mathsf{H}} \boldsymbol{h}_{m,n} \right\|^2. \tag{17}$$

Therefore the normalizing factor can be adapted from (15) and (16) as

$$\eta_n = \frac{\kappa_m}{\phi_m^2} \left\| \boldsymbol{b}_n^{\mathsf{H}} \boldsymbol{h}_{m,n} \right\|^2 \leq \frac{P_{\text{sat}} \left\| \boldsymbol{b}_n^{\mathsf{H}} \boldsymbol{h}_{m,n} \right\|^2}{\phi_m^2}. \tag{18}$$

To make the most use of transmission power and suppress the noise, the normalizing factor should take the upper bound as

$$\eta_n = \min_{m \in M_n} \frac{P_{\text{sat}} \left\| \boldsymbol{b}_n^{\mathsf{H}} \boldsymbol{h}_{m,n} \right\|^2}{\phi_m^2}. \tag{19}$$

The MSE is thus given as

$$\begin{aligned} \mathsf{MSE}(\widehat{p}_n, p_n; M_n, \boldsymbol{b}_n) &= \frac{\|\boldsymbol{b}_n\|^2 \sigma^2}{\eta_n} \\ &= \frac{\sigma^2}{P_{\text{sat}}} \max_{m \in M_n} \phi_m^2 \frac{\|\boldsymbol{b}_n\|^2}{\left\| \boldsymbol{b}_n^{\mathsf{H}} \boldsymbol{h}_{m,n} \right\|^2}. \end{aligned} \tag{20}$$

III. Problem Formulation

In this section, we model the target of the asynchronous FL framework and the MSE constraint caused by AirComp into a joint problem. Initially, we tackle this challenge through an exhaustive search for the optimal solution but this demands high computational complexity. To mitigate this complexity, we introduce a heuristic algorithm that combines DFS and BFS algorithms, incorporating subtree pruning. Specifically, the DFS algorithm is responsible for crafting the optimal linkage scheme in each epoch, while the BFS algorithm focuses on determining the beamforming vectors for HAPs.

A. Problem Definition

According to the asynchronous FL framework presented in Section II-A, a critical factor affecting the performance of global model $z^{[t]}$ is the volume of data

$$\boldsymbol{D}^{[t]} = (|d_1|, |d_2|, \cdots, |d_m|, \cdots, |d_M|)^\top \tag{21}$$

used to train local models. In the subsequent part of this paper, the notation $[t]$ is omitted for simplicity and $\boldsymbol{D}$ will represent $\boldsymbol{D}^{[t]}$. The best choice is to aggregate all local models in each epoch, which, however, is hard to be realized for the limitation caused by AirComp as presented in Section II-B. Since not all satellites are linkable to HAPs, the optimization target is to

maximize the volume of total training data utilized by aggregated local models through adjusting the beamforming vectors of HAPs and satellites that participate in the aggregation in epoch t together, limited by MSE and power constraints as

$$\begin{aligned} \underset{\boldsymbol{S}\in\mathbb{R}^{N\times M},\boldsymbol{b}_n\in\mathbb{C}^L}{\text{maximize}} \ & \|\boldsymbol{SD}\|_1 \\ \text{subject to} \ & \frac{\sigma^2}{P_{\text{sat}}}\max_{m\in M_n} s_{n,m}\phi_m^2\frac{\|\boldsymbol{b}_n\|^2}{\left\|\boldsymbol{b}_n^{\mathsf{H}}\boldsymbol{h}_{m,n}\right\|^2}\le\gamma,\ \forall n, \\ & \left|b_l^{(n)}\right|^2\le\frac{P_{\text{HAP}}}{|L|},\ \forall l, \end{aligned} \tag{22}$$

where linkage indicator $s_{n,m}\in\boldsymbol{S}$ is defined as

$$s_{n,m}=\begin{cases}1, & \text{if satellite } m \text{ transmits to HAP } n\\ 0, & \text{otherwise.}\end{cases} \tag{23}$$

with constraint

$$\sum_{n\in N} s_{n,m}\le 1,\ \forall M \tag{24}$$

and $\gamma>0$ is the MSE threshold for global model aggregation.

Unfortunately, solving the mixed combinatorial optimization problem (22) is challenging due to the presence of the combinatorial objective function $\|\boldsymbol{SD}\|_1$ and the nonconvex MSE constraint involving both the combinatorial variable $\boldsymbol{S}$ and the continuous variable $\boldsymbol{b}_n$. It is worth noting that the proposed transceiver design, which leverages AirComp, relies on having perfect channel state information (CSI) [36]. However, transmitting CSI feedback can result in significant overhead [37]. To mitigate this, an alternative approach is to perform the transceiver design at the HAP by solving problem (22) and computing equation (14), thus requiring only the channel state information at the HAP [38]. Once the transmit scalars are calculated, the HAP can provide feedback to each satellite by transmitting the corresponding scalar κ_m. Channel training for estimating CSI at the HAP can be achieved by transmitting pilot sequences from each satellite [39]. The feedback problem can be addressed using either unquantized analog feedback or quantized digital feedback [40].

B. DFS Algorithm for Linkage Scheme

To achieve the goal in (22), the first step is to generate the candidates of linkage scheme $\boldsymbol{S}$ that maximize $\|\boldsymbol{SD}\|_1$ and then validate if there exist beamforming vectors for HAPs to support this scheme. Here the linkage scheme $\boldsymbol{S}$ is generated by the DFS algorithm with subtree pruning as interpreted in 2. Initially, a perfect binary tree $\mathcal{T}$ with $M+1$ layers is constructed to record the participation of satellites M in uploading local models. At layer 0, namely the root layer, there is only one node

$$\mathcal{T}_{0,0}=0. \tag{25}$$

At each subsequent layer m, the nodes are constructed based on the nodes from the previous layer $m-1$ as

$$\mathcal{T}_{m,i}=\begin{cases}\mathcal{T}_{m-1,i/2}+d_m & \text{if } i \text{ is even (left child)}\\ \mathcal{T}_{m-1,(i-1)/2} & \text{if } i \text{ is odd (right child),}\end{cases} \tag{26}$$

where the left child of each node represents the situation where satellite m participates in global aggregation, and the right child represents the situation where it does not. The value of node $\mathcal{T}_{m,i}$ is the sum of the data volume of the participating satellites in $1,2,\ldots,m$ and thus the value of node $\mathcal{T}_{M,i}$ is the sum of data volume of all satellites in the $i-$th situation.

The DFS algorithm iterates over the binary tree $\mathcal{T}$ from its root $\mathcal{T}_{0,0}$ and keeps visiting the child nodes $\mathcal{T}_{m+1,2i}$ and $\mathcal{T}_{m+1,2i+1}$ of the current node $\mathcal{T}_{m,i}$ if the recorded maximum data volume D_{max} is lower than the possible one as

$$\mathcal{T}_{m,i}+\sum_{j=m+1}^{M} d_j > D_{max}. \tag{27}$$

The visiting condition (27) prunes the impossible subtree and shrinks the search space effectively. When the algorithm visits the nodes in the last layer $M+1$, it compares their values $\mathcal{M}_{m,i}$ with D_{max} to update the recorded maximum value and corresponding satellites participation scheme $\mathcal{P}$. Based on the final returned $\mathcal{P}$, the candidate linkage schemes can be generated with the constraint (24).

To further decrease the search space of the above DFS algorithm, some elements in $\boldsymbol{S}$ can be fixed to dispose of the impossible linkage schemes. The constraint of MSE

$$\frac{\sigma^2}{P_{\text{sat}}}\max_{m\in M_n} s_{n,m}\phi_m^2\frac{\|\boldsymbol{b}_n\|^2}{\left\|\boldsymbol{b}_n^{\mathsf{H}}\boldsymbol{h}_{m,n}\right\|^2}\le\gamma,\ \forall n \tag{28}$$

equivalently means that for any linkage $s_{n,m}=1$ between satellite m and HAP n, it satisfies

$$\frac{\sigma^2}{P_{\text{sat}}}\phi_m^2\frac{\|\boldsymbol{b}_n\|^2}{\left\|\boldsymbol{b}_n^{\mathsf{H}}\boldsymbol{h}_{m,n}\right\|^2}\le\gamma \tag{29}$$

which can be rewritten as

$$\frac{\sigma^2}{P_{\text{sat}}}\phi_m^2\|\boldsymbol{b}_n\|^2\le\gamma\left\|\boldsymbol{b}_n^{\mathsf{H}}\boldsymbol{h}_{m,n}\right\|^2. \tag{30}$$

Based on the compatibility of matrix norm multiplication, it follows that

$$\left\|\boldsymbol{b}_n^{\mathsf{H}}\boldsymbol{h}_{m,n}\right\|^2\le\left\|\boldsymbol{b}_n^{\mathsf{H}}\right\|^2\|\boldsymbol{h}_{m,n}\|^2. \tag{31}$$

For $\left\|\boldsymbol{b}_n^{\mathsf{H}}\right\|^2=\|\boldsymbol{b}_n\|^2$ and $\|\boldsymbol{b}_n\|^2\in\mathbb{R}$ and $\|\boldsymbol{b}_n\|^2\neq 0$, they can be removed from both sides of the inequality as

$$\frac{\sigma^2}{P_{\text{sat}}}\phi_m^2\le\gamma\|\boldsymbol{h}_{m,n}\|^2. \tag{32}$$

For any linkage that does not satisfy this condition, its state $s_{n,m}$ should be 0. Based on this property, some situations can be fixed and therefore decrease the number of linkage schemes. For the satellite m that can not link to all HAPs N in this epoch, namely

$$\sum_{n\in N} s_{n,m}=0, \tag{33}$$

it will be removed from the candidate satellites and thus the binary tree $\mathcal{T}$ will have one less layer for each infeasible satellite. Besides, these infeasible linkages decrease the number of

Algorithm 2 DFS Algorithm for Linkage Scheme

Initialize Feasible Linkage Matrix $\boldsymbol{F} = \mathbf{1}_{N\times M}$, $\mathsf{M}_{\text{data}} = 0$, participation scheme $\mathcal{P}$
function DFS(node=$\mathcal{T}_{m,i}$, layer=m)
  if $m = M$
    if $\mathcal{T}_{m,i}$.value $> \mathsf{M}_{\text{data}}$
      $\mathsf{M}_{\text{data}} = \mathcal{T}_{m,i}$.value
      $\mathcal{P} = \mathcal{T}_{m,i}$.path
    end if
  else
    if $\mathcal{T}_{m,i}$.value $+ \sum_{i=m+1}^{M} d_i > \mathsf{M}_{\text{data}}$
      DFS($\mathcal{T}_{m+1,2i}, \mathcal{T}_{m+1,2i+1}$)
    end if
  end if
end function
$\boldsymbol{F} \leftarrow$ ValidateLinkage($\boldsymbol{F}$, $\boldsymbol{h}_{m,n}$) by constraint (32)
$M \leftarrow M \setminus \{m\}$ in condition (33)
Construct tree $\mathcal{T}$ with $M+1$ layers by equation (26)
repeat
  Get participation scheme $\mathcal{P} \leftarrow$DFS($\mathcal{T}_{0,0}$)
  Get linkage schemes set $\boldsymbol{C_S}$ with $\mathcal{P}$ and $\boldsymbol{F}$
  Validate the feasibility of $\boldsymbol{S} \in \boldsymbol{C_S}$ with Algorithm 3
  $\mathcal{T} \leftarrow$PruneSubtree($\mathcal{T}, \mathcal{P}$)
until feasible linkage scheme $\boldsymbol{S}$ is generated
return Linkage scheme $\boldsymbol{S}$

candidate linkage schemes to be validated in the next step as well. If all of them are validated to be unachievable, the maximum subtree containing the child node $\mathcal{T}_{M,i}$ corresponding to this participation scheme $\mathcal{P}$ will be pruned from the binary tree $\mathcal{T}$ and the next participation scheme $\mathcal{P}'$ is generated at the base of the modified tree. The details of the DFS algorithm to search candidate linkage schemes are given in Algorithm 2.

In terms of computation complexity, to search the target linkage scheme exhaustively, all candidates should be listed and ordered by the sum of the data volume $||\boldsymbol{SD}||_1$. The number of candidates is 2^M because there are two states for each satellite and the number of satellites in the scenario is M. The computation complexity of the sorting algorithm is $\mathcal{O}(2^{M^2})$ for a list including 2^M elements. Therefore, the complexity of the exhaustive search algorithm is $\mathcal{O}(4^M)$.

Regarding the proposed DFS algorithm for the linkage scheme, the first step is to construct the $(M+1)$-layer complete binary tree with $2^{M+1}-1$ nodes. Then the complexity of locating the largest potential linkage scheme is $\mathcal{O}(M)$ and in the worst case of validating all candidates the number of whom is 2^M, the complexity is $\mathcal{O}(2^M M)$. Therefore, the complexity of the DFS algorithm is $\mathcal{O}(2^M(M+2))$. The comparison results of the complexity between the proposed DFS algorithm and the exhaustive search are shown in Fig. 3.

Besides, because the complete binary tree constructed in the DFS algorithm contains all candidate linkage schemes, which are all checked in order until the solution is found, the convergence of the DFS algorithm is self-evident.

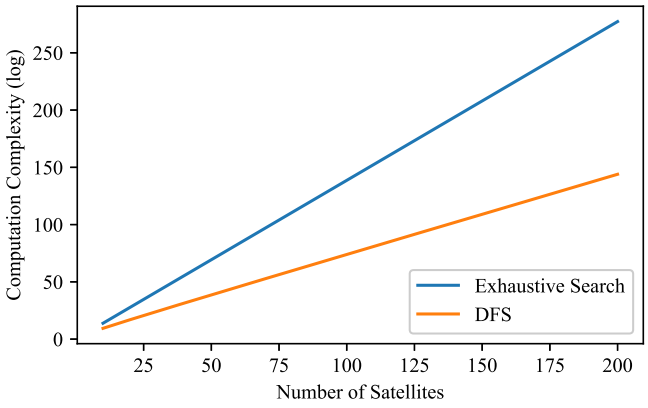


Fig. 3. The comparison between the complexity of the proposed DFS algorithm and the exhaustive search. Because of the huge differences between them, the results are shown in the form of log.

C. Solving Beamforming Vector

With the list of linkage schemes generated in Section III-B, the original problem (22) is turned into

$$\begin{aligned} &\text{find } \boldsymbol{b}_n \\ &\text{subject to } \frac{\sigma^2}{P_{\text{sat}}} s_{n,m}\phi_m^2 \frac{\|\boldsymbol{b}_n\|^2}{\left\|\boldsymbol{b}_n^{\mathsf{H}}\boldsymbol{h}_{m,n}\right\|^2} \le \gamma, \forall m, \\ &\left|b_l^{(n)}\right|^2 \le \frac{P_{\text{HAP}}}{|L|}, \ \forall l, \end{aligned} \tag{34}$$

which is still nonconvex for channel vectors $\boldsymbol{h}_{n,m}$ is independent of each other and change of any element in $\boldsymbol{b}_n$ leads to distinct results in different linkages. Therefore, we need to relax (34) through introducing constraint

$$\left|b_l^{(n)}\right|^2 = \frac{P_{\text{HAP}}}{|L|}, \ \forall l \tag{35}$$

which indicates that antenna arrays work at full power. In this way, the power of antennas at HAP n is fixed as

$$\sum_{l\in L}\left|b_l^{(n)}\right|^2 = \|\boldsymbol{b}_n\|^2 = |L|\frac{P_{\text{HAP}}}{|L|} = P_{\text{HAP}}. \tag{36}$$

And the problem (34) is relaxed to

$$\begin{aligned} &\text{find } \boldsymbol{b}_n \\ &\text{subject to } \frac{\sigma^2}{P_{\text{sat}}} s_{n,m}\phi_m^2 \frac{P_{\text{HAP}}}{\left\|\boldsymbol{b}_n^{\mathsf{H}}\boldsymbol{h}_{m,n}\right\|^2} \le \gamma, \forall m, \\ &\left|b_l^{(n)}\right|^2 = \frac{P_{\text{HAP}}}{|L|}, \ \forall l. \end{aligned} \tag{37}$$

Now then, it is accessible to iterate over the codebook to pick the appropriate beamforming vector that supports the linkage scheme generated in Section III-B. If no beamforming vector fulfils the MSE constraints of the best scheme, the next vector is validated, and the process repeats until all HAPs find their respective beamforming vectors, enabling communication with the selected satellites via AirComp. It is important to note that when we impose constraints on the amplification of elements within the beamforming vectors to reach their maximum values, the resulting solution is

suboptimal compared to the optimal one, which theoretically allows for variable amplification. Based on the relaxed energy constraint in (37), the BFS algorithm can be implemented to find the beamforming vector more efficiently instead of simply iterating over the modified codebook. The details are illustrated in Section III-D.

D. BFS Algorithm for Beamforming Vector

In problem (37), the target function can be regarded as looking for the beamforming vector $\boldsymbol{b}_n$ that enlarges the L2 norm so that the MSE constraint

$$\frac{1}{\left\|\boldsymbol{b}_n^{\mathsf{H}}\boldsymbol{h}_{m,n}\right\|^2} \leq \gamma_{m,n}, \forall m \tag{38}$$

is satisfied for all linkage $s_{n,m}=1$ and $\gamma_{m,n}$ is the notation for simplification as

$$\gamma_{m,n} = \frac{P_{\text{sat}}}{P_{\text{HAP}}\sigma^2\phi_m^2}\cdot\gamma. \tag{39}$$

The BFS algorithm is implemented to swiftly find out the $\boldsymbol{b}_n$ as the global optimal solution to problem (37) for the function

$$f(\theta_1,\ldots,\theta_L) = \left\|(e^{j\theta_1},\ldots,e^{j\theta_L})^{\mathsf{H}}\boldsymbol{h}_{m,n}\right\|^2 \tag{40}$$

is convex and the proof is shown in appendix A. Initially, following the energy constraint in (37), the subset codebook $\boldsymbol{W'} \subsetneqq \boldsymbol{W}$ is created as

$$\boldsymbol{W'} = \left\{ W_{a,p} \in \boldsymbol{W} \middle| |W_{a,p}|^2 = \frac{P_{\text{HAP}}}{|L|} \right\}. \tag{41}$$

Based on the definition of the codebook $\boldsymbol{W}$ in (10), the codebook $\boldsymbol{W'}$ can be equally defined as

$$\begin{aligned}\boldsymbol{W'} &= \{W_{1,1},\ldots,W_{1,p},\ldots,W_{1,P}\} \\ &= \left\{e^{j\frac{2\pi}{P}},\ldots,e^{j\frac{2p\pi}{P}},\ldots,e^{j\frac{2P\pi}{P}}\right\}.\end{aligned} \tag{42}$$

For the BFS algorithm, an undirected cycle graph $\mathcal{G}$ is constructed as

$$\mathcal{G} = (\boldsymbol{W'}, \boldsymbol{E}), \tag{43}$$

where the codewords $W_{1,p} \in \boldsymbol{W'}$ are nodes and edges in $\boldsymbol{E}$ only exist between adjacent nodes as

$$\boldsymbol{E} = \{(W_{1,1},W_{1,2}),\ldots,(W_{1,p},W_{1,p+1}),\ldots,(W_{1,P},W_{1,1})\}. \tag{44}$$

In the beginning, a beamforming vector $\boldsymbol{b}_n$ is generated as the start vector because the elements in it have the closest phases with those in the vector $\overline{\boldsymbol{h}_n}$ calculated as the sum of selected channel coefficients:

$$\overline{\boldsymbol{h}_n} = \sum_{m=1}^{M} s_{n,m}\boldsymbol{h}_{m,n}, \tag{45}$$

and thus the beamforming vector $\boldsymbol{b}_n$ is the solution to

$$\boldsymbol{b}_n = \arg\sup_{b_l^{(n)}\in\boldsymbol{W'}} \left\|\boldsymbol{b}_n{}^{\mathsf{H}}\overline{\boldsymbol{h}_n}\right\|^2. \tag{46}$$

In this way, the start vector $\boldsymbol{v}$ can support most linkages $s_{n,m}=1$ of HAP n to meet the MSE constraint (38) in just one calculation. Then it is added into the queue Q that records the vectors waiting to be visited and validated. In the validation step, the algorithm locates the first linkage $s_{n,m}=1$ that does not meet the MSE constraint (38) and substitutes each element $v_l \in \boldsymbol{v}$ to form the new vector

$$\boldsymbol{v'} = [v_1,\ldots,v_{l-1},v_{nbr},v_{l+1},\ldots,v_L] \tag{47}$$

with its adjacent nodes $v_{nbr} \in \mathcal{N}(v_l)$ in graph $\mathcal{G}$. If the new vector $\boldsymbol{v'}$ supports

$$\left\|\boldsymbol{v'}^{\mathsf{H}}\boldsymbol{h}_{m,n}\right\|^2 > \left\|\boldsymbol{v}^{\mathsf{H}}\boldsymbol{h}_{m,n}\right\|^2 \tag{48}$$

and does not lead any other linkage $s_{n,i}=1$ to violate its MSE constraint which is originally satisfied as

$$\frac{1}{\|\boldsymbol{v}^{\mathsf{H}}\boldsymbol{h}_{i,n}\|^2} \leq \gamma_{i,n} < \frac{1}{\left\|\boldsymbol{v'}^{\mathsf{H}}\boldsymbol{h}_{i,n}\right\|^2}, \tag{49}$$

it will be added to the queue Q and returned as the solution of beamforming vector $\boldsymbol{b}_n$ to the linkage scheme $\boldsymbol{S}$ in the case that it supports the MSE conditions of all linkages to HAP n as (38). The implementation of the BFS algorithm in solving beamforming vector is given in Algorithm 3.

In terms of computation complexity, to search the target beamforming vector exhaustively, all candidates should be validated in the worst case whose complexity is calculated as $\mathcal{O}(P^L)$ where P is the number of phases in the codebook as defined in (10) and L is the number of antennas on a HAP.

Algorithm 3 BFS Algorithm for Beamforming Vector

Require: Graph $\mathcal{G}$, Linkage scheme $\boldsymbol{S}$,
Channel coefficients $[\boldsymbol{h}_{1,n},\boldsymbol{h}_{2,n},\ldots,\boldsymbol{h}_{M,n}]$,
MSE thresholds $[\gamma_{1,n},\gamma_{2,n},\ldots,\gamma_{M,n}]$,
Create an empty queue Q
Create a set $visited$ to store visited nodes
Calculate $\overline{\boldsymbol{h}_n} = \sum_{m=1}^{M} s_{n,m}\boldsymbol{h}_{m,n}$
Calculate $\boldsymbol{v} = \arg\sup_{v_l\in\mathcal{G}} \left\|\boldsymbol{v}^{\mathsf{H}}\overline{\boldsymbol{h}_n}\right\|^2$
Enqueue $\boldsymbol{v}$ into Q
Add $\boldsymbol{v}$ to $visited$
while $Q \neq \varnothing$ **do**
  Dequeue a vector $\boldsymbol{v}$ from Q
  Locate the satellite m that $\frac{1}{\|\boldsymbol{v}^{\mathsf{H}}\boldsymbol{h}_{m,n}\|^2} \leq \gamma_{m,n}$
  for each element $v_l \in \boldsymbol{v}$ **do**
    for each neighbours $v_{nbr} \in \mathcal{N}(v_l)$ **do**
      $\boldsymbol{v'} = [v_1,\ldots,v_{l-1},v_{nbr},v_{l+1},\ldots,v_L]$
      if $\boldsymbol{v'} \notin visited$ **and** $\left\|\boldsymbol{v'}^{\mathsf{H}}\boldsymbol{h}_{m,n}\right\|^2 > \left\|\boldsymbol{v}^{\mathsf{H}}\boldsymbol{h}_{m,n}\right\|^2$ **then**
        Add $\boldsymbol{v'}$ to $visited$
        if $\boldsymbol{v'}$ does not violate the constraint (49) **then**
          Enqueue $\boldsymbol{v'}$ into Q
        end if
        if $\boldsymbol{v'}$ meet the constraint (38) **then**
          return $\boldsymbol{v'}$ is beamforming vector of HAP n
        end if
      end if
    end for
  end for
end while
return The linkage scheme $\boldsymbol{S}$ is unfeasible

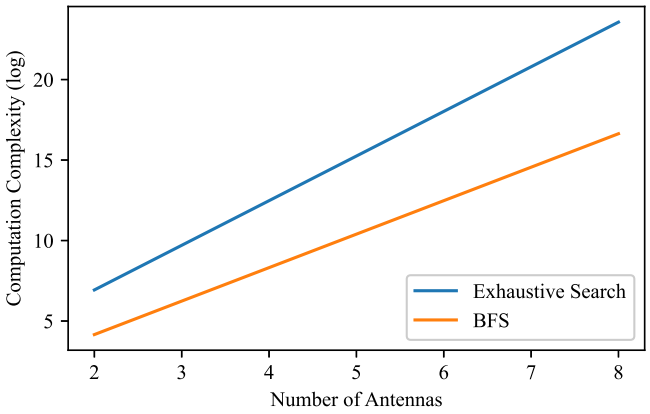


Fig. 4. The comparison between the complexity of the proposed BFS algorithm and the exhaustive search. Because of the huge differences between them, the results are shown in the form of log.

In our proposed BFS algorithm, the number of the phases that each antenna has to validate is halved based on the convex function (40). Therefore, its complexity in the worst case is calculated as $\mathcal{O}(\frac{P}{2}^{L})$. The comparison results of the complexity between the proposed BFS algorithm and the exhaustive search are shown in Fig. 4.

Besides, because the graph constructed in the BFS algorithm contains all candidate linkage schemes, which are all checked in order while the unavailable ones are removed at the same time before finding the solution, the convergence of the BFS algorithm is self-evident.

TABLE I
PARAMETERS OF SIMULATION SCENARIOS

Parameters	Values
Number of satellites	1-20
Number of HAPs	1-10
Altitude of HAPs	22km
Local training time (including time to collect data)	1 hour per epoch
Waiting time of retransmission	6 minutes
Local training epoch	1
Global training epoch	50
Volume of training data	0.7-1 gigabyte (GB) per epoch per satellite
Energy consumption of HAP	5 kilojoules (kJ) per reception
Size of per sample	104 kilobyte (KB)
Number of parameters of a deep learning model	2.88×10^8
Number of CPU cores	4
CPU clock frequency	1.43 gigahertz (GHz)
Transmission power	60 decibel-milliwatts (dBm)
Antenna gains of satellite and HAP	6.98 decibel-isotropic (dBi)
Noise temperature	354.81 Kelvin (K)
Bandwidth	0.5 GHz
Wavelength	15 millimeters (mm)
Power of an antenna	1.73 watts (W)

IV. Simulation Results

In this section, we conduct simulations to compare the proposed asynchronous FL framework and AirComp technique with synchronous FL framework and traditional computation-communication-separated situations. The proposed quantity-quality jointed linkage search algorithm is tested first to generate those satellites participating in the global model aggregation in each epoch and then the schedule is input into the federated learning simulation platform to get the deep learning models and verify the effect of the framework.

A. Simulation Setup

LEO Network and HAP. In the simulations, we set an LEO satellite network involving 20 satellites whose orbits are the same as those of Starlink. 10 HAPs are evenly spread at the equator, floating above the surface of the earth at an altitude of 22 km. In the scenarios where the number of HAPs is less than 10, the rest of HAPs still stay at the original locations instead of being relocated and equally spread at the equator so that the rest of them keep networking together to aggregate global models. Orbits of satellites can be described in two-line element (TLE) sets [41] so that HAPs can easily track the locations of satellites in a period long enough for them to communicate with satellites and update their TLE sets. The parameters of simulation scenarios are listed in Table I.

Channel Coefficient and Beamforming Codebook. For simplicity, we do not utilize a complex mathematical model to construct channels [42]. Instead, we assume the amplifications of channel coefficients are Rician distribution whose scale is inversely proportional to the square of the distance between HAP and satellite while the phase of each channel coefficient is uniformly scattered from 0 to 2π [43]. For each element in the beamforming vector, it has four amplification patterns distributing from 0.25 to 1 and eight phase patterns ranging from 0 to 2π.

*Asynchronous and Synchronous FL Framework.*The asynchronous FL framework is compared to the synchronous FL approach and AirComp is not implemented in some situations to evaluate its capability to converge the global model. The situations where AirComp is not used are indicated as LocalComp in the following content since computation is separated from communication and completed locally. In the synchronous FL, the aggregation procedure does not stop until all satellites upload their models. In the situations without AirComp, HAPs have to receive local models one by one. Regardless of computation heterogeneity, the time to collect data and train local models is set as 1 hour in both FL frameworks and the time to transmit models is omitted. This figure is calculated with the basic computation capacity and communication condition parameters in Table I. The calculation is demonstrated in detail in appendix B After satellites complete training, HAPs in asynchronous FL only collect models once from each satellite according to the linkage scheme while those in synchronous FL keep trying building

TABLE II
HYPERPARAMETERS OF DEEP LEARNING MODELS

Hyperparameters	Values
Model	MLP
Learning rate	0.05
Input dimension	784
Layers	[512, 256, 256, 128]
Output dimension	10
Activation	Linear rectification function (ReLU) [45]
Optimizer	Stochastic gradient descent (SGD) [45]
Momentum	0.5
Global epochs	50
Local epochs	1
Local batch size	10
Classes per satellite	10
Data type	IID

linkage with satellites until all of them transmit their models. Therefore, in the simulation, the time of one epoch is fixed as 1 hour in asynchronous FL and that in synchronous FL increases by 6 minutes with each attempt to communicate with satellites that have not uploaded their models. The data volume utilized to train local models for each satellite in each epoch is uniformly distributed from 0.7 to 1 GB. [1]

Energy Consumption. In terms of energy, we only focus on the energy consumed by HAPs to receive models, assumed to consume 1 kJ when the antenna arrays of a HAP work at full power [44]. Comparison between the optimal and suboptimal solution is presented in the simulation as well while in the latter situation the amplification of elements within the beamforming vectors is fixed as maximum values.

Deep Learning. To verify the effect of our proposed asynchronous FL framework, we use FedAvg as the synchronous FL framework and modify it into the asynchronous FL framework. A multilayer perceptron (MLP) classifier [46] is trained on top of a Bidirectional Long Short-Term Memory (Bi-LSTM) encoder [47], which includes 128 hidden units and two hidden layers, each with size 512. The Modified National Institute of Standards and Technology database (MNIST) [16] is a widely used collection of hand-written digit images, consisting of 60,000 training examples and 10,000 testing examples and it is utilized in the simulations. Besides, the EuroSAT dataset is utilized as well, showing the possibility of applying the asynchronous FL framework to realistic geospatial artificial intelligence tasks. It is a benchmark dataset used for land use and land cover classification tasks in the field of remote sensing and machine learning, including 10 different classes and 27,000 labelled remote sensing images in total. Both datasets involve 10 classes and are split into independent identically distribution (IID) data [48] based on the number of satellites in each simulation. The hyperparameters of deep learning models are listed in Table II.

[1]Here the value of the data volume refers to the data volume used for the calculation of throughput and energy efficiency in Section IV-B, C and D.

B. Throughput of Training Data over the Number of HAPs

The research [49], [50] has shown that the performance of deep learning models is related to the volume of training data. That is because as the amount of training data increases, deep learning models generally exhibit enhanced performance due to their ability to learn more comprehensive and nuanced representations of the input data. This principle holds particular importance in the context of FL, where multiple local models are trained on distributed data subsets and then aggregated to form a global model. Therefore, more training data learned by the local models that have been aggregated into the global model in a certain time can be regarded as a necessary indicator in evaluating the effect of the FL framework. Here the indicator throughput TP^t at epoch t is defined as

$$TP^t = \frac{\sum_{i=1}^{t}\sum_{m\in M^{[i]}}|d_m|}{\sum_{i=1}^{t}\Delta T_i}, \tag{50}$$

where the numerator indicates the sum of data volume utilized by all aggregated models from the very beginning to the present epoch t and the denominator refers to the total running time of the FL task. ΔT_i refers to the time of global epoch i.

In this simulation, a scenario is constructed with 20 satellites and the MSE threshold $\gamma = 15$dB and Fig. 5 illustrates the throughput of different FL frameworks with the increase of the number of HAPs. When there are only a few HAPs, satellites in the synchronous FL can hardly upload their local models to the HAP and therefore the epoch duration is delayed, resulting in low throughput. With the increase of HAPs, this problem is alleviated but it cannot be thoroughly solved until all satellites can link to HAPs as soon as they finish training local models in all global epochs. This requires not only lots of HAPs that cover the earth but also the proper MSE threshold that supports linkages over a far distance.

Compared to the synchronous FL, the asynchronous FL is more robust to changes in the number of HAPs because even if satellites fail to upload their local models in the present epoch, they can keep training and send models to any HAP eventually. Thus the throughput of the asynchronous FL framework should approach a constant when the global epoch is large enough

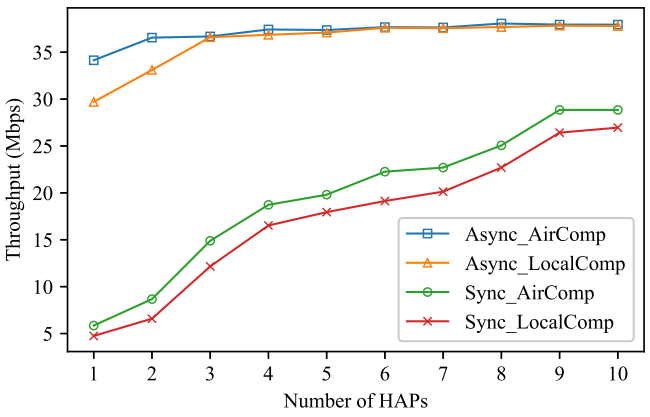


Fig. 5. Throughput of different FL frameworks over the number of HAPs.

TABLE III
AVERAGE THROUGHPUT OF DIFFERENT FRAMEWORKS

Framework	Average Throughput (Mbps)	Imporvement over Synch & LocalComp
Async & AirComp	37.14	114.34%
Async & LocalComp	36.18	108.80%
Sync & AirComp	19.57	12.90%
Sync & LocalComp	17.33	0%

TABLE IV
AVERAGE ENERGY EFFICIENCY OF DIFFERENT FRAMEWORKS

Framework	Average Energy Efficiency (GB/kJ)	Imporvement over Synch & LocalComp
Asyn & AirComp	1.16	581.77%
Asyn & LocalComp	0.76	345.29%
Syn & AirComp	0.25	48.83%
Syn & LocalComp	0.17	0%

despite various HAPs. Since the AirComp technique supports HAPs to accept multiple models at the same time, it decreases the epoch duration of synchronous FL and helps HAPs in asynchronous FL to link to more satellites in one epoch, which leads to higher throughput in both FL frameworks.

Table III lists the mean values of throughput for four frameworks in all scenarios, and that of asynchronous FL with AirComp is 114.34% higher than that of synchronous FL without AirComp, while the asynchronous FL improves it by 108.80% and the AirComp technique boosts it by 12.90%.

C. Energy Efficiency

To evaluate the energy consumption of FL frameworks, energy efficiency E^t at epoch t is defined as the ratio between the data volume applied to aggregated local models and the sum of the energy ΔE_i consumed by all HAPs at global epoch i for receiving local models:

$$E^t = \frac{\sum_{i=1}^{t} \sum_{m \in M^{[i]}} |d_m|}{\sum_{i=1}^{t} \Delta E_i}. \tag{51}$$

Similar to the simulations in Section IV-B, those in this section set the scenarios with 20 satellites and the MSE threshold $\gamma = 15$dB as well. Fig. 6 illustrates the changes in energy efficiency with the number of HAPs in different frameworks.

With only one HAP in the scenario, asynchronous FL shows higher energy efficiency compared to synchronous FL. This is because, in this case, satellites in both frameworks can barely connect to the HAP, and those in asynchronous FL accumulate training data and upload local models only once, instead of transmitting models in every epoch as in synchronous FL. As the number of HAPs increases, satellites in asynchronous FL have more opportunities to upload models, causing the energy efficiency to decline and stabilize at a constant value, which is still higher than that of synchronous FL.

The AirComp technique enables HAPs to receive local models from several satellites at the same time and therefore it constantly improves the energy efficiency of both FL frameworks unless there are so few satellites in the scenarios that AirComp never helps HAPs receive multiple models simultaneously. When there are sufficient HAPs and the MSE threshold is high enough to support all satellites to upload their local models immediately after they complete training, asynchronous and synchronous FL frameworks share the same energy efficiency because their aggregation schedules are identical.

Table IV lists the mean values of energy efficiency for four frameworks in all scenarios, and that of asynchronous FL with AirComp technique is 581.77% higher than that of synchronous FL without AirComp technique, while the asynchronous FL enhances it by 345.29% and the AirComp technique boosts it by 48.83%.

D. Throughput of Training Data over MSE Threshold

To evaluate the performance of frameworks in various communication environments, the simulation over MSE thresholds is conducted and throughput is selected as the measurement of the accuracy of global deep learning models. Fig. 7 displays the throughput of an FL framework with 20 satellites and 10 HAPs at the fiftieth hour after starting the FL process. It

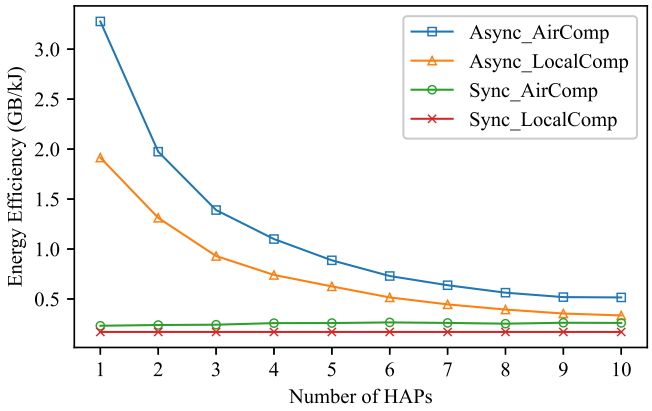


Fig. 6. Energy efficiency of different FL frameworks over the number of HAPs.

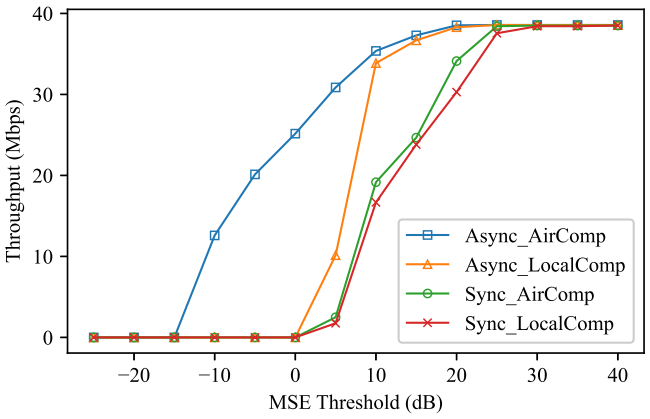
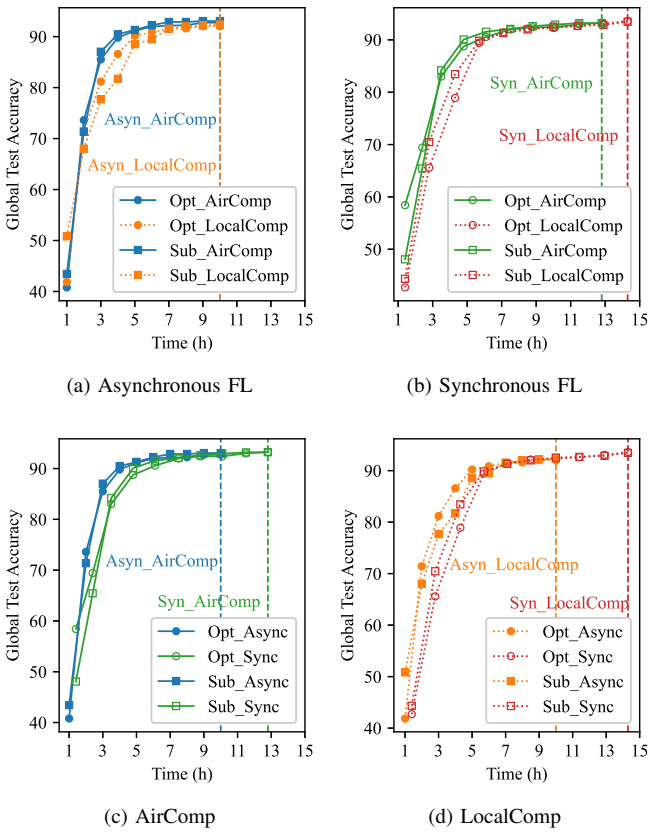


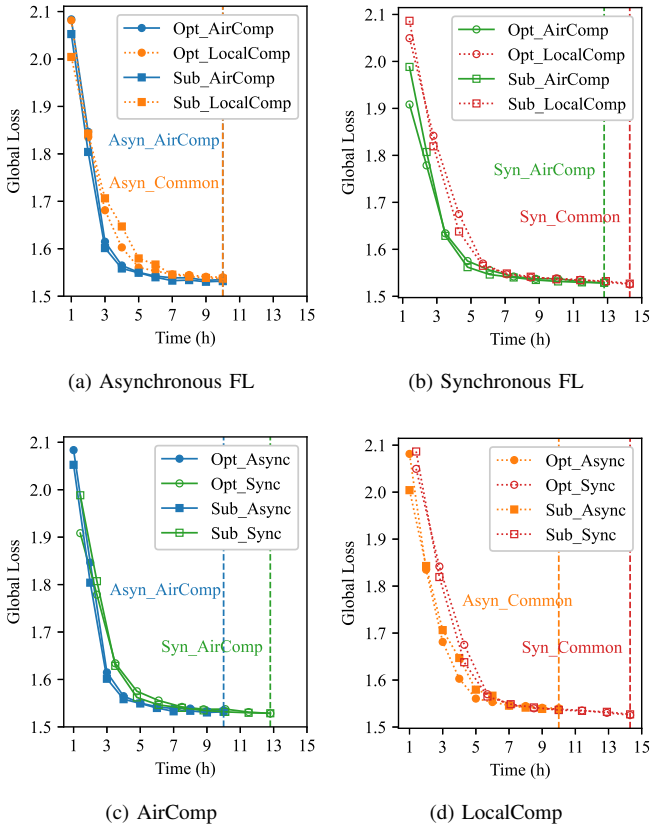
Fig. 7. Throughput of training data over MSE thresholds.



(a) Asynchronous FL (b) Synchronous FL

(c) AirComp (d) LocalComp

Fig. 8. Comparisons of accuracy curves in different situations. All of them contain ten epochs and the dotted lines indicate the tenth epoch. The models are trained on the MNIST dataset.



(a) Asynchronous FL (b) Synchronous FL

(c) AirComp (d) LocalComp

Fig. 9. Comparisons of loss curves in different situations. All of them contain ten epochs and the dotted lines indicate the tenth epoch. The models are trained on the MNIST dataset.

exposes that the throughput of asynchronous FL overtops that of synchronous in most situations since the former one enables satellites to sum up the training data volume till they upload models in fixed epoch duration while the latter one has to lengthen the aggregation time. In extreme circumstances where the MSE threshold is incredibly high, the asynchronous FL is no longer distinct from the synchronous FL because satellites are feasible to HAPs at all times, and vice versa.

E. Performance of Federated Learning

MNIST Dataset. Fig. 8 involves the accuracy curves of the global models trained in various frameworks but the same scenario. These models are trained on the MNIST dataset. In all scenarios, 20 satellites and 10 HAPs exist in the FL framework and the MSE threshold is set as $\gamma = 15$dB. According to our observation, the declines of loss values are less than 0.2% for all situations at the tenth epoch. Therefore we consider that models in all situations converge at the tenth epoch and only the accuracy and loss curves of global models in the first 10 epochs are plotted.

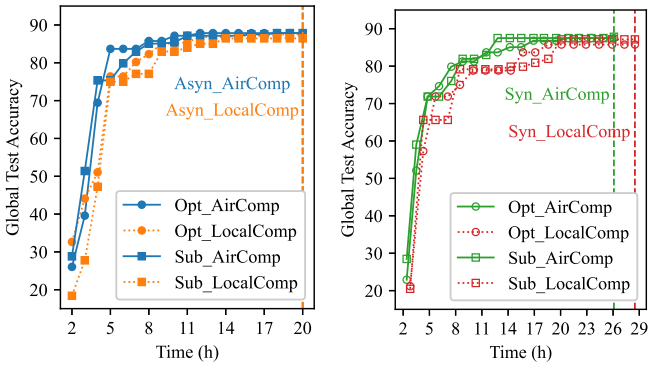
In Fig. 8a and 8b, the AirComp technique improves the convergence rate of global models for the FL frameworks utilize it to achieve apparently higher accuracy from the second hour to the sixth hour. Besides, curves in Fig. 8c and 8d reveal the fact that asynchronous FL helps the global model to be more precise at the beginning epoch in less time. In the last three subfigures, the frameworks using AirComp or asynchronous FL obviously shorten the duration of each epoch

TABLE V
PERFORMANCE OF MODELS

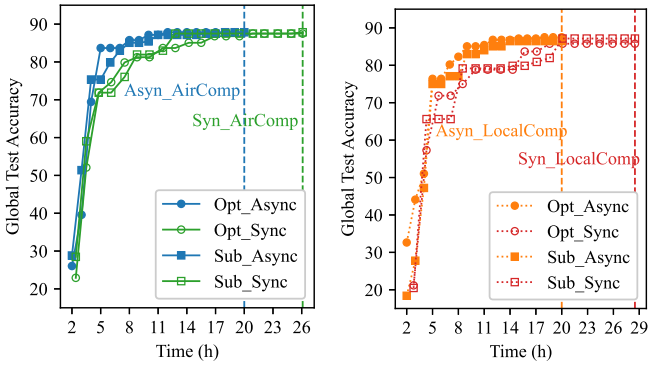
Frameworks	Time (hour)		Comparison of Time	
	5 HAPs	10 HAPs	5 HAPs	10 HAPs
Asynch & AirComp	4.14	3.93	46.05%	61.70%
Synch & AirComp	7.21	4.79	80.20%	75.20%
Asynch & LocalComp	6.93	4.97	77.09%	78.02%
Synch & LocalComp	8.99	6.37	100%	100%

and the convergence time of global models. Compared to the synchronous FL framework that does not utilize AirComp, the convergence time cost by the synchronous FL framework with AirComp is 10.49% less than its. In the scenarios where AirComp is used, the asynchronous FL framework decreases the convergence time by 21.88%. If the asynchronous FL framework and AirComp are both implemented, the convergence time will fall by 30.07% in contrast to the synchronous one without AirComp. In all FL frameworks, the performance of global models generated by the suboptimal aggregation schedule is extremely close to those generated by the optimal one. Fig. 9 demonstrates the loss curves which show similar phenomena as those in the accuracy curves.

Table V records the time required for the global models in four different frameworks to reach an accuracy of 90%. From the table, it is evident that the asynchronous FL with AirComp significantly reduces the time needed for model convergence compared to the other frameworks. Specifically, with 5 HAPs,

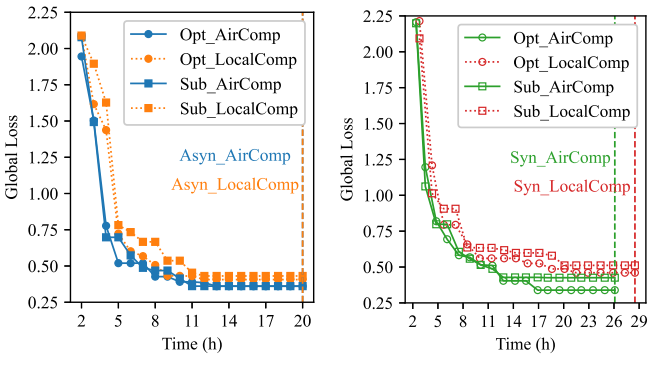


(a) Asynchronous FL (b) Synchronous FL

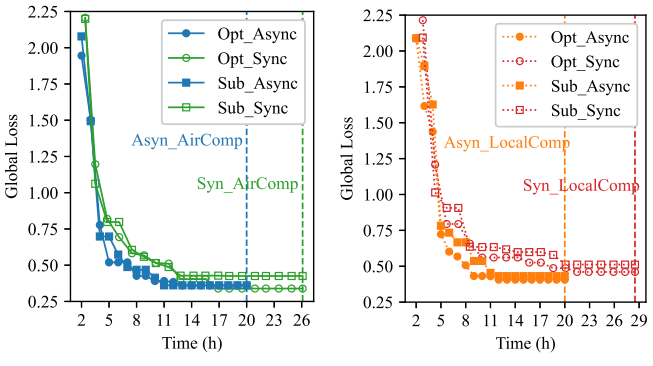


(c) AirComp (d) LocalComp

Fig. 10. Comparisons of accuracy curves in different situations. All of them contain ten epochs and the dotted lines indicate the tenth epoch. The models are trained on the EuroSAT dataset.



(a) Asynchronous FL (b) Synchronous FL



(c) AirComp (d) LocalComp

Fig. 11. Comparisons of loss curves in different situations. All of them contain ten epochs and the dotted lines indicate the tenth epoch. The models are trained on the EuroSAT dataset.

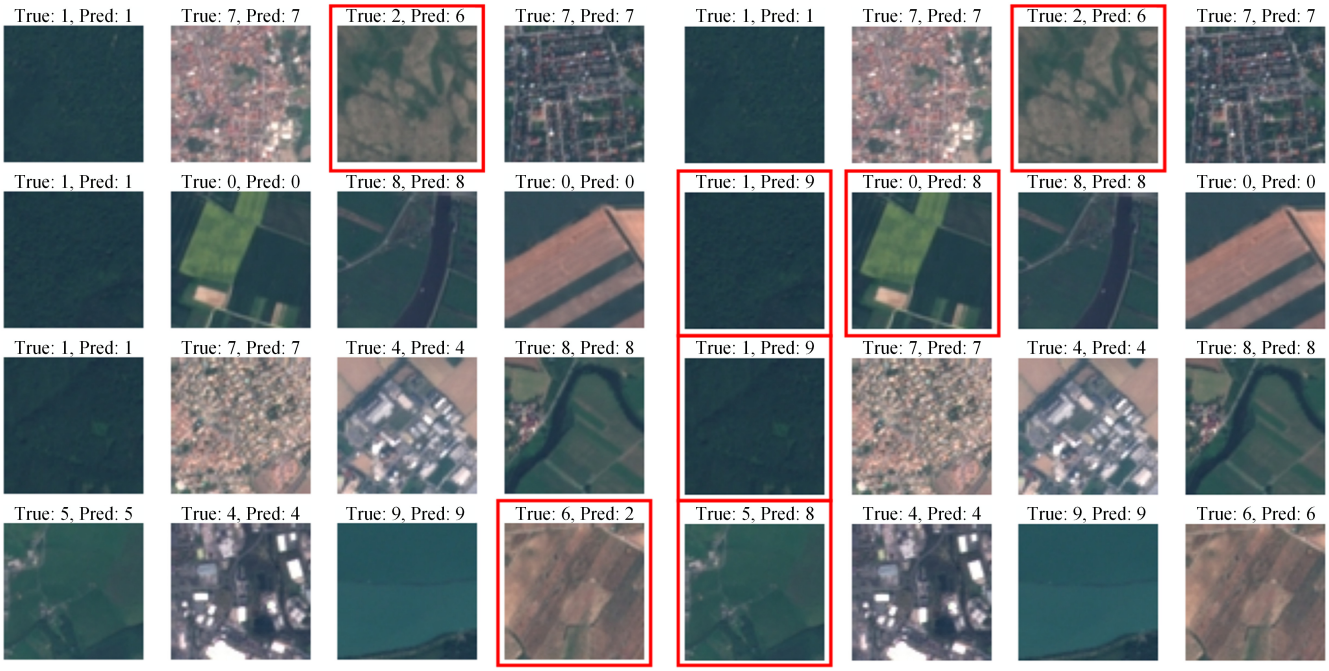
the asynchronous FL with AirComp takes only 4.14 hours to achieve the target accuracy, which is approximately 46.05% of the time required by the synchronous FL without AirComp, the slowest framework in this comparison. Similarly, with 10 HAPs, the time required is reduced to 3.93 hours, which is about 61.70% of the time needed by the synchronous FL without AirComp. These results highlight the efficiency of the asynchronous FL with AirComp in accelerating the training process, thereby achieving faster convergence in both cases.

EuroSAT Dataset. Fig. 10 and Fig. 11 involves the accuracy and loss curves of the global models in various frameworks but the same scenario. These models are trained on the EuroSAT dataset. These curves show the similar trends to those of the models trained on the MNIST dataset, accompanied with a little difference brought by the EuroSAT dataset. The convergency time of these models is longer and the final accuracies of these models are lower because the remote sensing images in the EuroSAT dataset are harder to learn. The asynchronous FL framework improves the convergency rate of the global model, from 22 hours to 12 hours as shown in Fig. 10d. In all situations, the asynchronous FL framework and AirComp raise the performance of the models at the early stage.

Fig. 12 shows the prediction results of the deep learning models trained in asynchronous FL framework with AirComp (Fig. 12a) and synchronous FL framework without AirComp (Fig. 12b) at the tenth hour. When the asynchronous FL finishes the tenth global epoch, the synchronous FL just finishes the fifth global epoch. The instances marked by red boxes are predicted wrong and the mistakes in Fig. 12b (five) are obviously more than those in Fig. 12a (two). This indicates that at the tenth hour in our simulation, the model in asynchronous FL framework can achieve the requirement of accuracy but the model in synchronous FL framework cannot.

Future Work. In the simulations with real deep learning tasks, some problems in asynchronous FL framework based on the AirComp technique appear to be solved, involving:

- *Non-IID Data:* In reality, satellites usually use non-IID data to train local models and this may lead to weight divergence. How to modify the present FL framework to prevent this problem may introduce new variables into the optimization function in this paper and interesting challenges as well.
- *Staleness Issue:* In asynchronous FL, the stale model problem is mutual concerned and someone solves it by not only adjusting the weight coefficients of aggregating models but discarding toxic models as well. The latter approach may result in conflict to the target of more training data volume and how to balance it is challenging.
- *Uncertain CSI:* This paper assumes the perfect CSI between satellites and HAPs. It would be interesting to solve the problem in an uncertain channel during model aggregation by predicting the channel coefficient with a deep learning model.
- *Adaptive Power:* This work assumes the antennas to work at full power to reduce computation complexity, but how



(a) Asynchronous FL with AirComp (b) Synchronous FL without AirComp

Fig. 12. Comparisons of the prediction results of land cover classification tasks. At the tenth hour, the model trained in asynchronous FL framework with AirComp accomplishes the tenth global epoch and the model trained in synchronous FL framework without AirComp accomplishes the sixth global epoch. The red boxes mark the error results.

to calculate the power adapted to the CSI is a challenge.

V. Conclusion

In this paper, we proposed an asynchronous federated learning framework, based on the over-the-air computation technique, applied to LEO satellite network for rapid global model aggregation and high convergence rate. This framework utilizes high-altitude platforms as a core network to receive local models, integrate them into global models, and disperse global models to satellites. To balance the demand of the federated learning framework for more training data volume with that of the over-the-air computation technique for lower signal distortion, we proposed a model aggregation scheduling algorithm based on depth-first search and breadth-first search algorithm with subtree pruning. We first removed infeasible linkage between satellites and high-altitude platforms and then generated the linkage scheme by the depth-first search algorithm. Next, at the cost of energy, we assumed the antennas work at full power to find the beamforming vectors of high-altitude platforms by the breadth-first search algorithm at less computation complexity. Simulation results demonstrated the superiority of the asynchronous federated learning framework in throughput and energy efficiency in various scenarios. The global model accuracies suggested that the suboptimal aggregation schedule worked out by our proposed algorithm shared a similar capability with the global optimal one.

Appendix A
Appendix A: Proof of the Convex Function

To prove that the function (40) is convex, we need to show that its Hessian matrix is positive semidefinite. The function can be rewritten as

$$f(\boldsymbol{\theta}) = \left\| \boldsymbol{h}_{m,n}^{\mathsf{H}} \boldsymbol{v} \right\|^2 \tag{52}$$

where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_L)$, $\boldsymbol{v} = (e^{j\theta_1}, \ldots, e^{j\theta_L})$, and $\boldsymbol{h}_{m,n}$ is a given vector. The Hessian matrix H of $f(\boldsymbol{\theta})$ is given by

$$H = \frac{\partial^2 f(\boldsymbol{\theta})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^{\mathsf{T}}}. \tag{53}$$

Let

$$\boldsymbol{v}_i = (0, \ldots, 0, \underbrace{e^{j\theta_i}}_{i\text{-th element}}, 0, \ldots, 0)^{\mathsf{T}}, \tag{54}$$

then

$$\frac{\partial \boldsymbol{v}}{\partial \theta_i} = \frac{\partial}{\partial \theta_i} \left(e^{j\theta_i} \right) = j e^{j\theta_i} \tag{55}$$

and

$$\begin{aligned} \frac{\partial^2 f(\boldsymbol{\theta})}{\partial \theta_i \partial \theta_j} &= \frac{\partial}{\partial \theta_j} \left(\boldsymbol{h}_{m,n}^{\mathsf{H}} j e^{j\theta_i} \right) \\ &= j^2 \boldsymbol{h}_{m,n}^{\mathsf{H}} e^{j\theta_i} e^{-j\theta_j} = -\boldsymbol{h}_{m,n}^{\mathsf{H}} e^{j(\theta_j - \theta_i)}. \end{aligned} \tag{56}$$

Thus, the Hessian matrix H is:

$$H = -\boldsymbol{h}_{m,n}^{\mathsf{H}} \begin{bmatrix} e^{j(\theta_1-\theta_1)} & e^{j(\theta_2-\theta_1)} & \ldots & e^{j(\theta_L-\theta_1)} \\ e^{j(\theta_1-\theta_2)} & e^{j(\theta_2-\theta_2)} & \ldots & e^{j(\theta_L-\theta_2)} \\ \vdots & \vdots & \ddots & \vdots \\ e^{j(\theta_1-\theta_L)} & e^{j(\theta_2-\theta_L)} & \ldots & e^{j(\theta_L-\theta_L)} \end{bmatrix}, \tag{57}$$

namely

$$H = -\boldsymbol{h}_{m,n}^{\mathsf{H}} \begin{bmatrix} 1 & e^{j(\theta_2-\theta_1)} & \dots & e^{j(\theta_L-\theta_1)} \\ e^{j(\theta_1-\theta_2)} & 1 & \dots & e^{j(\theta_L-\theta_2)} \\ \vdots & \vdots & \ddots & \vdots \\ e^{j(\theta_1-\theta_L)} & e^{j(\theta_2-\theta_L)} & \dots & 1 \end{bmatrix}. \tag{58}$$

The matrix above is a circulant matrix, and it's known that circulant matrices are diagonalized by the Discrete Fourier Transform (DFT) matrix F, such that $F^{\mathsf{H}}AF = \Lambda$, where Λ is a diagonal matrix containing the eigenvalues of A. Since the eigenvalues of a circulant matrix are the DFT of its first row, Λ is diagonal with elements $\lambda_k = \sum_{i=1}^{L} e^{j(\theta_i-\theta_1)} e^{-j2\pi(k-1)(i-1)/L}$.

Since the diagonal elements of Λ are the sums of the first row of the circulant matrix, they are all real and nonnegative. Thus, H is negative semidefinite, proving that the function $f(\boldsymbol{\theta})$ is convex.

APPENDIX B
APPENDIX B: DETAILS ABOUT SIMULATION SETTINGS

The values of the parameters in the following are given in Table VI. For the satellite m, if it collects 1 GB data for training ($|d_m|$), the number of samples c_{sample} is

$$c_{sample} = \lceil \frac{|d_m|}{s_{sample}} \rceil = \lceil \frac{1 \times 10^6}{104} \rceil = 9615 \tag{59}$$

where s_{sample} indicates the size of the each sample. Normally, a parameter in a deep learning model requires 2 FLOPs (Floating Point Operations) in forward propagation (one multiplication and one addition) and 4 FLOPs in backward propagation (two multiplications and two additions). Given the size of the deep learning model G, the total FLOPs F_{FLOPs} required in training can be calculated as

$$\begin{aligned} F_{\text{FLOPs}} &= 6 \times \mathsf{G} \times c_{sample} \\ &= 6 \times 2.88 \times 10^8 \times 9615 \\ &= 1.661 \times 10^{13}. \end{aligned} \tag{60}$$

Based on the number of CPU cores C_m^{CPU} and CPU clock frequency f_m, the number of FLOPs that the CPU executes per second f_{FLOPs} is

$$f_{\text{FLOPs}} = C_m^{CPU} \times f_m = 4 \times 1.43 \times 10^9 = 5.72 \times 10^9. \tag{61}$$

In this way, the time of a local epoch t_{train} is

$$t_{\text{train}} = \frac{F_{\text{FLOPs}}}{f_{\text{FLOPs}}} = \frac{1.661 \times 10^{13}}{5.72 \times 10^9} = 2904.55 \tag{62}$$

whose unit is second.

In terms of communication time, the maximum data rate R should be calculated as

$$R = B \cdot log_2(1 + SNR) \tag{63}$$

where B is bandwidth and SNR refers to the signal-to-noise ratio (SNR). SNR is calculated with the transmission power P_{trans}, the antenna gains of satellite G_m and HAP G_{HAP}, the noise temperature T_{temp}, the bandwidth B_{bw} the Boltzmann constant K_B and the free-space path loss $\mathcal{L}$ as

$$SNR = \frac{P_{\text{trans}} G_m G_{\text{HAP}}}{T_{\text{temp}} B_{\text{bw}} K_B \mathcal{L}}. \tag{64}$$

The path loss is determined by

$$\mathcal{L} = (\frac{4\pi d_{\text{dist}}}{\lambda_{\text{wl}}})^2 \tag{65}$$

where d_{dist} is the minimum distance that allows for LoS communication between a satellite and a HAP, and λ_{wl} is the wavelength of the transmitted signal. According to the values given in the table, the data rate R can be calculated as nearly 16 Mbps with the positions of satellites and HAPs assumed in our simulations. In this way, the time to transmit a deep learning model t_{trans} can be calculated as

$$t_{\text{trans}} = \frac{4 \times 8 \times \mathsf{G}}{R} = \frac{4 \times 8 \times 2.88 \times 10^8}{16 \times 10^6} = 576 \tag{66}$$

because a parameter in the model is regarded as a float and requires 4 bytes, namely 32 bits. 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. Since the local epoch is set as 1, the time for a global epoch of the asynchronous FL framework is the sum of the training time t_{train} and transmission time t_{trans}, near to 3480 seconds. Considering the overhead during this process t_{overhead}, the time of a global epoch in asynchronous FL framework t_{epoch} is defined as

$$\begin{aligned} t_{\text{epoch}} &= t_{\text{train}} + t_{\text{trans}} + t_{\text{overhead}} \\ &= 2904.55 + 576 + 119.45 \\ &= 3600, \end{aligned} \tag{67}$$

namely 1 hour.

TABLE VI
ADDITIONAL PARAMETERS FOR SIMULTAIONS

Parameter	Value
Volume of training data $\|d_m\|$	1 GB
Size of per sample s_{sample}	104 KB
Number of parameters of a deep learning model G	2.88×10^8
Number of CPU cores C_m^{CPU}	4
CPU clock frequency f_m	1.43 GHz
Transmission power P_{trans}	60 dBm
Antenna gains of satellite G_m and HAP G_{HAP}	6.98 dBi
Noise temperature T_{temp}	354.81 K
Bandwidth B_{bw}	0.5 GHz
Wavelength λ_{wl}	15 mm
Time for training t_{train}	2904.55 seconds
Time for transmission t_{trans}	576 seconds
Time of overhead t_{overhead}	119.45 seconds
Time of a global epoch t_{epoch}	1 hour
Power of an antenna P_{ant}	1.73 W
Energy consumption for receiving a model E_{rec}	5 kJ

In terms of energy consumption, it is highly related to the power of the receiver and that is too complex to calculate comprehensively in our manuscript. Therefore, here we simply assume the power for each antenna P_{ant} is 1.73 W. Given the time of transmission t_{trans}, the energy consumption of a receiver E_{rec} for receiving a model can be calculated as

$$\begin{aligned} E_{\text{rec}} &= |L| \times P_{\text{ant}} \times t_{\text{trans}} \\ &= 5 \times 1.73 \times 576 \\ &= 4982.4 \end{aligned} \tag{68}$$

which is approximated to 5 kJ for simplification.

REFERENCES

[1] P. Yue *et al.*, "Low earth orbit satellite security and reliability: Issues, solutions, and the road ahead," *IEEE Commun. Surv. Tutorials*, vol. 25, no. 3, pp. 1604–1652, thirdquarter 2023.

[2] X. Cao *et al.*, "Edge-assisted multi-layer offloading optimization of LEO satellite-terrestrial integrated networks," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 2, pp. 381–398, Feb. 2023.

[3] H. Xv *et al.*, "Joint beam scheduling and beamforming design for cooperative positioning in multi-beam LEO satellite networks," *IEEE Trans. Veh. Technol.*, vol. 73, no. 4, pp. 5276–5287, Apr. 2024.

[4] J. Zhu, Y. Sun, and M. Peng, "Timing advance estimation in low earth orbit satellite networks," *IEEE Trans. Veh. Technol.*, vol. 73, no. 3, pp. 4366–4382, Mar. 2024.

[5] N. Guo, L. Liu, and X. Zhong, "Task-aware distributed inter-layer topology optimization method in resource-limited LEO-LEO satellite networks," *IEEE Transactions on Wireless Communications*, vol. 23, no. 4, pp. 3572–3585, Sept. 2024.

[6] A. S. Antonarakis and A. Guizar Coutiño, "Regional carbon predictions in a temperate forest using satellite lidar," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 10, no. 11, pp. 4954–4960, Nov. 2017.

[7] 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.

[8] 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.

[9] 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.

[10] K. Wei *et al.*, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 3454–3469, Apr. 2020.

[11] M. Zhao *et al.*, "IC2S-Swarm: When digital twin meets collaborative ISR," *IEEE Commun. Mag. (under review)*, Mar. 2024.

[12] H. Zhang and L. Hanzo, "Federated learning assisted multi-UAV networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 14 104–14 109, Nov. 2020.

[13] Y. Wang *et al.*, "Federated learning over LEO satellite," in *2022 IEEE GC Wkshps*, Dec. 2022, pp. 1652–1657.

[14] Z. Zhai *et al.*, "FedLEO: An offloading-assisted decentralized federated learning framework for low earth orbit satellite networks," *IEEE Trans. Mob. Comput.*, vol. 23, no. 5, pp. 5260–5279, May 2024.

[15] H. Wu and P. Wang, "Node selection toward faster convergence for federated learning on Non-IID data," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 5, pp. 3099–3111, Sept. 2022.

[16] Z. Zhao *et al.*, "Ensemble federated learning with Non-IID data in wireless networks," *IEEE Trans. Wireless Commun.*, vol. 23, no. 4, pp. 3557–3571, Apr. 2024.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] M. Elmahallawy and T. Luo, "AsyncFLEO: Asynchronous federated learning for LEO satellite constellations with high-altitude platforms," in *IEEE BigData 2022*, Dec. 2022, pp. 5478–5487.

[22] 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.

[23] 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.

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

[25] 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.

[26] 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.

[27] G. Zhu, Y. Wang, and K. Huang, "Broadband analog aggregation for low-latency federated edge learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 491–506, Jan. 2020.

[28] 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.

[29] M. Elmahallawy, T. Luo, and K. Ramadan, "Communication-efficient federated learning for LEO constellations integrated with HAPs using hybrid NOMA-OFDM," *IEEE J. Sel. Areas Commun.*, vol. 42, no. 5, pp. 1097–1114, May 2024.

[30] Q. Wei, X. Chen, and Y. F. Zhan, "Exploring implicit pilots for precise estimation of LEO satellite downlink doppler frequency," *IEEE Commun. Lett.*, vol. 24, no. 10, pp. 2270–2274, Oct. 2020.

[31] Z. Jia *et al.*, "Toward data collection and transmission in 6G space–air–ground integrated networks: Cooperative HAP and LEO satellite schemes," *IEEE Internet Things J.*, vol. 9, no. 13, pp. 10 516–10 528, Oct. 2022.

[32] 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.

[33] M. Zhao *et al.*, "Federated learning for 6G: A survey from perspective of integrated sensing, communication and computation," *ZTE Commun.*, vol. 21, no. 2, pp. 25–33, Jun. 2023.

[34] A. Dowhuszko and J. Hämäläinen, "Performance of transmit beamforming codebooks with separate amplitude and phase quantization," *IEEE Signal Process Lett.*, vol. 22, no. 7, pp. 813–817, Jul. 2015.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[40] 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.

[41] A. A. Spiridonov *et al.*, "Small satellite orbit determination methods based on the doppler measurements by Belarusian State University ground station," *IEEE J. Miniaturization Air Space Syst.*, vol. 2, no. 2, pp. 59–66, Jun. 2021.

[42] R. Yuan and M. Peng, "Single-input multiple-output scattering based optical communications using statical combining in turbulent channels," *IEEE Trans. Wireless Commun.*, vol. 23, no. 4, pp. 2560–2574, Apr. 2024.

[43] X. Zhang, M. Peng, and C. Liu, "Impacts of antenna downtilt and backhaul connectivity on the UAV-enabled heterogeneous networks," *IEEE Trans. Wireless Commun.*, vol. 22, no. 6, pp. 4057–4073, Jun. 2023.

[44] H. Jung, I. Lee, and J. Joung, "Security energy efficiency analysis of analog collaborative beamforming with stochastic virtual antenna array

of UAV swarm," *IEEE Trans. Veh. Technol.*, vol. 71, no. 8, pp. 8381–8397, Aug. 2022.
[45] Q. Yang *et al.*, "Learning two-layer ReLU networks is nearly as easy as learning linear classifiers on separable data," *IEEE Trans. Signal Process.*, vol. 69, pp. 4416–4427, Jul. 2021.
[46] J. Tang, C. Deng, and G. Huang, "Extreme learning machine for multilayer perceptron," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 27, no. 4, pp. 809–821, Apr. 2016.
[47] Y. He, P. Wang, and Q. Zhu, "Improved Bi-LSTM with distributed nonlinear extensions and parallel inputs for soft sensing," *IEEE Trans. Ind. Inf.*, vol. 20, no. 3, pp. 3748–3755, Mar. 2024.
[48] 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.
[49] 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.
[50] J. Kaplan *et al.*, "Scaling laws for neural language models," *arXiv preprint arXiv:2001.08361*, Jan. 2020.

Yansong Huang received his B.S. degree in Electronic Engineering from Beijing University of Posts and Telecommunications, China in 2022. He is pursuing his master's degree in Information and Communication Engineering at Beijing University of Posts and Telecommunications. His research interests include federated learning and integrated sensing, communication and computation in 6G.

Xuan Li (Senior Member, IEEE) has been an Associate Professor at Beijing University of Posts and Telecommunications, China, since 2021. She received the B.Eng. degree in optical information science and technology from the Beijing Institute of Technology, China, in 2012, and the Ph.D. degree with Southampton Wireless, University of Southampton (UoS), U.K., in 2016. She worked as a Post-Doctoral Research Fellow with UoS, and then a Senior Data Scientist with Trilateral Research Ltd., UK, from 2017 to 2021. Her research interests include AI-driven networks, UAV swarm network management, and human-swarm interaction.

Moke Zhao received her B.S. degree in Electronic Information Engineering from Beijing University of Posts and Telecommunications, China in 2022. She is pursuing her master's degree in Communications Engineering at Beijing University of Posts and Telecommunications. Her research interests include edge computing and wireless communication in 6G.

Haiyan Li is currently a Ph.D. student in Information and Communication Engineering at Beijing University of Posts and Telecommunications, Beijing, China. He received the M.S. degree in Computer Science and Technology from the National Innovation Institute of Defense Technology, Beijing, China, in 2023. Before that, he received the B.S. degree in Information Management and Information Systems from Jilin University, Jilin, China, in 2020. His current research interests include UAV clustering and resource allocation.

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.